

US 4 • AMIGA • PC • C64 • C128 • C16 PLUS 4 • AMIGA

YOUR

COMMODORE

AN ARGUS SPECIALIST PUBLICATION

July 1989 £3.50

PC GAMES

What should you buy?

JEFFY DOS

High Speed
C128/64 Disk access

STAR LC-10C

Best for Colour
Printing?

UNBEATABLE PROGRAMS:

• MouseBtl • Quiz Search • C64 Windows

REVIEWED:

• The Real Ghostbusters • Para Assault Course
• Demon 4 Winter • Las Vegas Casino • Time Thief

ISSN 0261-8777



0261-8777

CONTENTS

VOLUME 5
NUMBER 10



DANCE DANCE REVOLUTION



THE SIMS

ADULTS ONLY

4 GHOSTS 4 TIME
Score 101000



THE SIMS



THE SIMS

THE SIMS

THE SIMS

FEATURES

- | | | | |
|--|----|---|----|
| * Grubby's Day Out | 10 | * Demon's Winter | 42 |
| Surreal gaming fun | | A new experience in role-playing | |
| * Incredible Shrinking Sphere | 10 | * Las Vegas Casino | 42 |
| Escape from the 3-D maze | | Gambling has never been this dull | |
| * Le Mans | 11 | * Pan Assault Course | 44 |
| Speed is of the essence! | | A game that sorts out the race from the boys | |
| * The PC 10 | 12 | * Real Ghostbusters | 74 |
| Commodore's PC 10 is given the once over | | Save the world from ghosts and goblins | |
| * Graphics on the 64 | 15 | * Time Thief | 74 |
| Improve your graphic capabilities | | A chase across time | |
| * Jiffy DOS | 18 | * The Star LC10 | 60 |
| Speed up your 64 drive | | A closer look at the dot-matrix, six colour printer | |
| * PC Games | 34 | * Amiga Workbench | 76 |
| Ten top games for the Commodore PC | | Hot tips for Amiga owners | |

REGULARS

- | | | | |
|---|----|---|----|
| * Data Statements | 6 | * Extending Basic | 52 |
| More news from the world of Commodore | | How to add your own commands to Commodore Basic | |
| * 128 Corner | 28 | * Software for Sale | 78 |
| The first of a regular series for 128 users | | If you don't want to type it - buy it! | |
| * Sprite Library | 37 | * Back Page | 82 |
| More sprites for use in your own programs | | Corrections, Info and 'The Nibbles' | |

PROGRAMS

- | | | | |
|--|----|--|----|
| * Mouse 80 | 20 | * Windows on the 64 | 55 |
| Use a mouse on your 128 in 80-column text mode | | Windows on your C64 made easy | |
| * Quick Search | 30 | * Sprite Sprite | 62 |
| A handy Basic search routine | | Sprites for the Plus 4 - can it be possible? | |
| * Head For Home | 45 | | |
| Continues on last month's Plus 4 game | | | |

Editor: Stuart Cooke
 Deputy Editor:
 Paul Wilkinson
 Technical Editor: Paul Fox
 Group Editor:
 Mark Webb
 Advertisement Manager:
 Paul Ramage
 Ad-Copy Control:
 Ruth Taylor
 Artist: Alan Archer
 Designer: Neil Swenson
 Organisation: Zborni
 Printers: Chase Webb

Your Commodore magazine appearing on the first Friday of each month. Argus Specialist Publications Limited, Editorial and Advertisement office: Two Grosvenor Square House, Grosvenor Way, Strand, London WC2N 3ET. Telephone: (0442) 40550. Subscription rates upon application to Your Commodore Subscription Department, Editors Ltd, 2 River Park Estate, Bickington, Here, HP8 4HT, U.K.A. Subscription Agent: New City Worldwide Publications, 4044 West 238th Street, Torrance CA 90505 U.S.A.

ARGUS PRESS GROUP

The contents of this publication including all articles, designs, drawings and programs and all copyrights and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights are reserved by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction in whole or in part without the prior written consent of the Company. © 1988. Distribution SM Publications & Leighton Court Road, London SW16 3PS. Printed by Chase Webb, Plymouth. Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. Wherever effort is made to thoroughly check programs published, the errors we cannot be held responsible for any errors that do occur.

ISSN 0269-8277

ABC

THE MYTH COLUMN

For all members of the Official Secrets Magazine Society we about to give you a treat. From the writers of *The Firm*, *Corruption* and *The Guild Of Thieves* comes 'Myth'. Myth is a new comic strip, where you play the part of Poseidon, God of the Sea. Your task is not as easy as you think. Lurking somewhere is 'Helmet of Invisibility'. Your mission: Jim should

(Sorry, wrong program.) Your mission is to find it. A problem? Yes, no, maybe? How do you find something that is invisible? Enough of the plot, Jim and get it. 'Myth' will be made available initially for Allen XT, Amiga and Macintosh. Other formats will follow shortly (C64 and other 8-bit disk only machines).

Fourthly,
*Mythology Scrolls, 1 Chapel
 Court, London, SE1 1WW
 Tel: 01 433 6121/428*

Data Statements

THE WORLD IS INSECURE

Computer Security Ltd, Britain's largest systems security specialists, will be demonstrating its PC products and systems at the PC User Show (9-11 May at Olympia). They will be displaying stand number 924. There will be a variety of products on view, covering such topics as Hacking, Viruses, File Security, and Electronic Mail protection, etc, etc. For

anyone that values their data, this must be worth a visit.

Fourthly

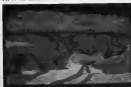
*Computer Security Ltd
 Oliver House, 18 Marsh
 Fields, London, EC2 1PL
 Telephone: (071) 672244*

JINN OF THE SPHINX

Innovative Software has commissioned exclusively for the House Computer Club, the fourth in its 3D game, Total Eclipse II, (The Sphinx Hunt). Your mission is to rebuild the Sphinx that was destroyed in the ancient revolution. You can only do that before the eclipse, which is just 1 hour away. There are 12 parts of the sphinx to collect from the underground network of tunnels and caverns. As an added bonus, the player with the highest score will win for themselves a trip to Hawaii to see a real Total Eclipse. Total Eclipse and Total Eclipse II are available together as a special presentation. The game is available for the House Computer Club only and costs £11.99 for both disks. The game will be released for the Spectrum, Amstrad CPC and C64.

Fourthly

*Innovative Software Ltd
 Zephyr One, Colindale Park,
 Aldenham, Bore, RG7
 4QH
 Tel: 07156 77381*



ON SCREEN READING

Following the major changes in the way that Prestel is organised four new on-line leisure magazines have been launched.

The magazines are a response to recent surveys detailing the users' areas of interest. Used now on screen magazines have been rather boring to look at, since updating a page always seemed to take ages. John Thomas, General Manager for Prestel Leisure has stated that "Reorganisation of the data base and new regional resources mean that reading the magazine will be as easy as flicking to your favourite page in a newspaper."

The magazines cater for those interested in the latest news, sport, leisure and games.

Newsday gives you up to the minute news, from the Observer, available 24 hours a day. The magazine will feature news headlines updated throughout the day. Financial news, and features Travel features, and weather reports, are provided by AA Roadwatch, and the Met office, while car and cycle will feel the instantly accessible British Rail timetables extremely useful.

Sportseye caters for the amateur sports-person, providing instant information for all major counties and first class matches, updated every five minutes, with daily news and commentary on every match. Football fans will find fixtures, match previews, results and league tables. Horse racing, motor racing, rugby, American football, golf and other special sporting events will be covered.

In a few days Sportseye gives subscribers a chance to interview their favourite sporting personalities.

Add to all of this quizzes, the ability to order souvenirs from the Sportseye and Sportseye, the Sports Centre's database of sports facilities

throughout the UK, and you've got what is probably the most comprehensive package available for sports fans everywhere.

Look? is designed to be an entertainment magazine covering travel, health, music, theatre, food and even an advice unit.

Information for Look? is provided by the Consumers Association, British Rail, AA Roadwatch, The National Theatre and Prestel's own Cordair Wine Guide.

Readers of Look? will be able to purchase wine, order tickets and take part in regular interviews with personalities.

For n' Games City is as no name suggests, dedicated to those who are looking for a little light relief.

The Pier has one-armed

bandits, role-shots and other games - with cash prizes. Prices of hi-fi equipment, videos, and records will all be given away to regular quizzers.

Prestel is an on-line database that can be accessed with a modem and a computer with relevant software. Subscription costs £1.00 plus VAT per quarter. An extra connection charge is also made of 1p per minute off-peak and 7p per minute peak.

Contact: Datalog UK, on 0442 337390, for more information.

KNIT & PURL 2

Here's a novel idea for all you knitters out there. You know how difficult it is to read those massive patterns? What with their long squiggly

and black and white forams? Well, help is now at hand. An enterprising chap from Bradford, Mr Harry Mann, has devised a program for transposing these patterns into a more easily readable form. At the moment, the program is available for the C64, but hopefully other formats will be entered for shortly to quote Harry, "This programme helps people who want to design their own garments to print out the patterns in a more easily readable form before they start to knit up."

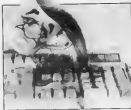
Touchline
Bradford Enterprise Service,
Cot. Hall Bridge Street,
Bradford W. Yorks BD1
197 Tel 0374 253399

Multilingual Keyboards

Do you wish that your computer keyboard sported symbols for French, Italian, Arabic or even Farsi? You do? Well, then you may be interested in the latest product from Kador.

Kador are now offering a range of keyboard kits for 19 foreign languages. The key-top kit consists of all the correct characters, diacritical marks, accents and punctuation symbols. The key-tops are made from rigid PVC and are simply stuck on to your existing keyboard. Prices start at £17.50.

As well as producing multilingual keytops Kador also produces a unique keyboard cover, SoftType.



SoftType is designed to fit exactly over each key on your computer keyboard making the keyboard safe from accidental spillages of tea, coffee etc. (if you had to remove spaghetti from someone's C64 FDs Prices are: C64/C16 £7.50 PC, A286, A386, A286 £18.95.

Kador can be contacted at: Unit 4, Pottersdown Industrial Estate, Abingdon, Oxon. OX14 3EP Tel: 0443 140261.

SOFT CHARITY

This summer, computer owners will be able to pick up some cheap software, and help a major charity at the same time.

OXFAM (Oxford's leading overseas aid and development charity, working on emergency relief, and long-term development projects in Africa, Asia, Latin America, and the Middle East, is asking for your software donations throughout June and July. The majority of the donated software will be sold at selected London shops during a special computer promotion month running on June 25th.

So, get checking your computer software. Do you really need that three-ferm up that you only play once a year? Now that you've

TWO FROM ONE

Strategic Studies Group (SSG) announces the release of two new titles in their ever growing array of programs. The first, *Gold of the Americas* is an historical strategy game. You know the kind set up, resources, trade, rival and potentially be master of everything you see and at the same time keep everyone happy. The second, *Pier King*, is an im-

mageded adventure. The world is in its worst state of darkness and chaos. During your journey through uncharted waters, ancient temples, and burning deserts, you fight off hordes.

Release date for these two are planned for the early summer. *Gold of the Americas*, will be available on PC format at £24.95 and *Pier King* will be for the C64 at £23.95.



Computer Graphics for Segments

By now you have all probably heard about the launch of yet another satellite TV service in the UK: French Satellite Broadcasting.

Incisive Software, producers of Percentage 3D graphics programs, have teamed up with BSB and Broadward, who produced the award-winning TV series *Knights of the Round Table*, to produce a 3D graphics presentation of *The Satellite Game*.

The game will place these youngsters inside a space shuttle to dash with English, an alien satellite, that is threatening to blow up the solar system.

The contestants have to construct and defend the core of the planet earlier by playing a droid-controlled Larry.

Freeplay from Incisive, will be used to give a true 3D representation of the world inside the satellite.

BSB is set to begin broadcasting in September 1989. These channels will be offered at first. Now, a sports and news service *Globe*, a general entertainment channel, and *The Movie Channel* which will offer 6 first-run feature films a week for a subscription of £9.99 per month. For more information contact BSB, 79 Southampton Road, London, SW7 1EN.

The Satellite Game can be found in January 1990



French Protection

The arguments about radiation, and other claims from monitor screens affecting the user, is one that seems to go on and on. One counter the manufacturers see there's no problem, then a different company comes up with a new solution to the low-radiation problem we've just been told doesn't exist.

For those who'd rather be safe than sorry, Aquila Permanent is a new entry into the protective market. The French manufacturer claims that the new screen filter offers a fully effective shield against low-level radiation. The filter also protects against static, screen glare, and screen reflections.

The Aquila permanent screen shield is designed to fit any monitor and is available in the UK from Acusdata.

An Alternative Label

Alternative Software is set to film yet another budget label into the pocket-money software world.

The first new program on the label, *Warrior*, will be available short-on up, War, which was a very popular game when it was sold at full price. Price point War is just £2.99 for the C64 version.



The Aquila Permanent protects from screen glare, reflections, static and low level radiation.

ALL TANGLED UP

A new era of role-playing adventure is about to burst forth. Orna are planning the release of a new role RPA entitled 'Tangled Tales', subtitled 'The Misadventures of a Wizard's Apprentice'. We are informed that this will be a light-hearted and comical approach to the usual fantasy RPA.

You are the cast in the role of a wizard's apprentice with three difficult tasks to perform. Stories consist of haunted houses, medieval tournaments and country lanes. The game features music and some cutscenes, none of the art graphics and an unusually good story line. *Tangled Tales* will be released for the C64 and IBM PCs and compatibles. As yet, no definite price has been announced.

Touchdown

Boxing, 3 Market Place, Fitchley, Glen, GL1 8DA Tel 0446 34336



Another Band License

Demark is continuing its series of block-buster related games with the computer version of the latest Band movie *License to Kill*.

License to Kill The Computer Game, closely follows the plot of the film. In the game you'll take control of 007 in a helicopter chase, an under-water scene and a race to the border as Bond tries to prevent the film badgie Sanchez escaping with a massive haul of drugs.

License to Kill will be released in June for the Amiga, PC and C64.

License to Kill for the Amiga and PC is £49.95. *License to Kill* for the C64 is £29.95.



The PC 10



Andrew Brown takes a closer look at the Commodore PC 10

The PC 10 is the cheapest machine in Commodore's range of PC-XT compatibles. If you designed the bargain business PC! This is an overcrowded market, so Commodore really needed to get it right for the machine to be successful. Inevitably, however, some corners have been cut. Our group's Commodore team have been using it as a C64 owner upgrading to 16-bit technology. These are people who have been loyal to Commodore for a long time, so the following comments are made with those particularly in mind.

The Hardware

The PC 10 is very much a standard PC, both in terms of looks and internal architecture. It sits on a

three-bay system, with the monitor sitting on top of the system unit and the keyboard connected by a cable and DIN plug on the right-hand side. The dimensions of the system unit are approximately 30" x 14.3" x 5.5". This means that the box is slightly deeper than its width, and as a little awkward to position in an ordinary desk.

The front panel overhangs slightly, though what I cannot touch, because the keyboard won't slide under it, being too high. The overhang contains a reset, and on the right side has the keyboard connector and reset switch, which felt very wobbly on the well-travelled review machine. Inspection of the rear of the machine revealed an array of ports: including parallel

serial, mouse, composite video and RGB Universal, a set of four DDP switches were also visible. Using these different monitor types can be selected without having to remove the outer case.

The mouse port allows connection of Commodore's own 1315 mouse without having to tie up valuable ports in expansion slots. The power switch is also at the back, just above the power-on socket. There was no power-on socket, so at least two plugs are needed for this system. The power supply itself is rated at 75 watts, so it should be sufficient to power anything plugged into the expansion slot. There are none of these, although the aluminium covers at the back suggest four. Given that almost every



yling is built into the main board that should be ample. All the slots are full length, so there should be no problems with bulky items such as hard disks on-a-card.

Externally, everything except the front grille is made of metal, and feels very sturdy. The finish may not be to your taste, being a drab shade of grey, but it will be very durable. Removing the case is just a matter of removing the six fixing screws, and sliding a self-locking wedge reveals the disk drive bays, of which there are three: two floppy and one hard. The latter mounts to the left of the floppies in a vertical position when present, and also accommodates an expansion slot by having the controller built onto the main circuit board.

The keyboard is of the now-standard 103 key enhanced type. This differs from the old 84 key type in having the function keys along the top, and a separate numeric unit. Models 500 and 2000 owners will be familiar with it already, as it's just an almost identical. Two new models, are a cable-tied, and legs that lock into position positively.

Foot's is a highly subjective matter, but I think most users will find the action of the keys perfectly adequate. Getting used to the layout will probably prove a problem, and left-handers like myself will not be so reassured of this layout. Indeed the review machine didn't have the correct system disks. Also the software needed to set the Keyboard up was not properly installed, resulting in some strange

things happening when keys were pressed. I suspect that if all machines are sent out in the same way, a lot of wasted telephone calls will result.

The Monitor supplied was the familiar 1084C monitor, so I won't comment on it here, except to suggest that a stereo monitor is overkill for a single PC. More interesting is the display adapter, which can support different kinds of monitors, and hence graphics standards. Colour monitors such as the 1084 will produce CGA graphics, which are more suitable for games. Monochrome monitors can display Hercules graphics, and give EGA25 text with crisp characters. It's not possible to have a dual purpose machine, with monitors for both serious and games applications connected at the same time.



Processor Speed

There is something of an obsession in PC circles, with endless quoting of clock speeds in reviews. However, anyone who can be bothered to understand that this has a direct bearing on how fast the machine is to use. The PC10 III uses the original 80486 processor, which is the slowest of the latest 80486 family of chips, but clocked at up to 9.54 Mhz. This is double the original speed, and gives very nearly twice the performance. Needless to say, performance is a great improvement on most if not machinery, but not in every way.

In use

Switching the machine on brings a storage bar to life, and as the display starts, a number of messages appear at the top of the screen. These are slightly irrelevant, and I wonder if a better course of action would be to screen them from a set up program. You then have to boot MS-Dos by entering a system disk. The process is very similar to running Workbench on an Amiga, but when it's done, all you get is just a plain A prompt. The machine arrives running at the 4.77Mhz (slow) processor speed as marked on the change speed, you have to press three keys at once, or use the speed slider supplied. This is just plain, silly, and could lose Commodore sales in the showroom. After all, in the land of places where this machine is competing, sales people are unlikely to have saved things out sufficiently to change it.

The PC10 will run all the software you'd expect - that is, everything having packages requiring a hard disk, or EISA standard graphics adapter. Surprisingly though, changing the processor's clock speed doesn't make much difference to the computer. On reflection however, it's not so odd, as the limiting factor on most computer programs is the speed of the operator, not the machine. Only the benchmarks really show the greater processing power achieved, but even benchmarks should be taken with a pinch of salt. In the end, a computer is a productivity tool, and whether the one you buy meets your needs is the only question that must be answered. As a power though, a PC-XT compatible, like the PC10, should be sufficient for most home and general use.



Documentation

As befits a plain vanilla computer, the PC10 comes with plain documentation. Three manuals are provided two of which (for MS-Dos and QWRTerm) are standard Microsoft fare. As such they are perfectly adequate, but not very friendly to new users. The third, and thickest volume is the Users Manual. This does attempt to make the introduction to MS-Dos a little less painful, but it is however, let down by the standard of production (featuring as it does many line drawings, and fuzzy photographs). Also the manual seems to be discussing low-level programming, something I would rather see put into a technical

reference manual. It is worth bearing in mind though, that after the initial set-up, most people will have no further need for this manual.

Conclusion

Comparing the Commodore PC10 with its nearest competitor the Amstrad PC1512, reveals few glaring differences in either. The Amstrad is cheaper, but the Commodore has a better display, and so on. It might be the end come down to who offers the better handling deal. At the moment Commodore are handling a greater the MIPS1200 - with every PC1000. That, in my book, just about swings it.



Graphics on the 64

*Part two of our introduction to Graphics programming
on the C64*

Last month, I began describing the manipulation of graphics on the 64. Due to the lack of space, however, I had to leave two important aspects unattended.

Firstly, there's the small matter of bit-mapped graphics. Last month, I described how you can redefine a character's shape. If you could fill the screen with 1,600 different individual characters, you could manipulate the whole display in any way you wanted.

Unfortunately, you only have access to 256 characters under normal circumstances. But fear not, the VIC chip offers a solution - bit 5 of register \$D011 allows the activation of a bitmap mode. Type in the following command and see what happens:

```
POKE $D010, POKE($D010) OR 32
```

You should then have a screen full of jumbled rubbish, the top half comprising lines and random dots, the bottom half comprising the dull characters on. Clear the screen and type some random characters. The character colour should change where each character is placed, and you should see a flashing cursor.

What you have now is a view of the block of ram from memory location 0 to \$100. The flickering at the top of the screen is the changing of new page locations by the operating system. The normal screen memory is used to provide the character and background colours, and the memory from 0 to \$100 defines the dot patterns.

If you want to use bit map mode, you'll need to use a more suitable memory area. The easiest option, if you're using small Basic programs and you don't want to move Basic, is to select the block of memory from \$100 to \$100. You do this by loading the memory register as before. Press RUN/STOP and RESTART to reset the display, and type in the following:

```
10 POKE $100,POKE($100) OR 32
20 POKE $100,POKE($100) CH 1
30 FOR I = 0 TO 7000
40 POKE $100+I,0
50 NEXT I
```

This time you get a more regular display, which is progressively cleared. Lines 30 to 50 clear the bit map by setting zero bytes. Replace line 40 with the following and see what happens:

```
40 POKE $100+I,AND(1)*255
```

You can see that by changing the memory block, you can change the bit map. As I said before, the screen memory defines the colours of the bitmap. The bottom four bits of each screen location specify the background colour and the top four bits define the character colour.

Imagine that you want a white background and a yellow foreground (the value for white is 1 binary 0001, the value for yellow is 7 binary 0111). Combine them, and we get \$110000 or 113. Add the following lines to the above program, and see the effect.

```
40 FOR I = 0 TO 8000
50 POKE $100+I,$113
60 NEXT I
```

Voilà! The required colour combination is obtained.

As before, two resolutions are available, depending on the value of bit 4 of register \$D010. To set high resolution bitmap with a horizontal resolution of 120 dots, you use:

```
POKE $D010,POKE($D010)AND255
```

To set multicolour bitmap you use:

```
POKE $D010,POKE($D010)OR16
```

Both modes have a vertical resolution of 200 dots. But how do we control which dot is set or cleared? Assume you want to set a dot with a horizontal position of X, and a vertical resolution of Y. The following subroutines will set this dot.

```
ROUTINE SETDOT:
  MOVEMV $A0,$A1
  ADDI $A0,X
  ADDI $A1,Y
  RTS
ROUTINE
```

To erase a dot replace line \$B000 with:

```
POKE POKE $B000,POKE $B000 AND NOT($113)
```

Type the above subroutines with the following program:

```
10 POKE $100,POKE($100) OR 32
20 POKE $100,POKE($100) CH 1
30 FOR I = 0 TO 7000
40 POKE $100+I,0
50 NEXT I
60 FOR I = 0 TO 1000
70 POKE $100+I,1
80 NEXT I
90 FOR X=0 TO 100
100 FOR Y=0 TO 200
110 SETDOT
120 END
```

This time line 70 sets the background colour to white, and the foreground colour to black. The program should draw a diagonal bar from the top left hand corner.

Multicolour mode uses the same dot per system as characters to determine which colours are used. The

colours are determined in the following way:

BIT PATTERNS	COLORS SOURCES
00	WHITE
01	Upper 4 bits of colour
	memory
10	Lower 4 bits of colour
	memory
11	Lower 4 bits of colour from

You will have noticed that bit map routines are very slow. Multicolour mode is even more so, until you learn to handle pairs of dots. For successful use of bitmap graphics, it's necessary to access assembly routines for drawing.

The final area needing refinement is the use of sprites. A sprite is a block of graphical information, rather like a character which can be placed anywhere on the screen. A sprite consists of a square block 24 dots wide and 21 characters high. Each row uses 3 bytes, so that each sprite requires 3×21 bytes of memory. The VIC chip has a set of registers which determine how sprites are handled. All you need to do is insert the correct values in the relevant registers.

First we must set up a pattern. You can draw a sprite in a memory anywhere so that used by characters, but it's a lot more difficult. It's better to use a sprite decoder. The sprite pointers are stored in blocks of 64 bytes, starting at memory location 0. Sprite pattern 0 occupies memory locations 0 to 63, pattern 1 occupies 64 to 127 and so on. You can use the following formula to determine the start address of sprite pattern X:

$$\text{address} = 64 \times \text{block} + 64X + 65536$$

Since the VIC chip is looking at a specific memory bank, this address is really offset from the start address of the bank. Since the default bank for base is bank 0, the bank start address is 0. Since the 64 is an eight bit machine only, 8 sprites are supported. The VIC chip finds the sprite pattern from a block of eight sprite pointers. These are located 1600 bytes after the start of the video memory. In the default set up, the pointers therefore start at 1024×1600 or 3948.

The pointers for sprite 0 is held in the first pointer, the pointers for sprite 1 in the second pointer, and so on. If you want to set up sprite 3 in the pattern held in location 12388, you set the sprite pointer as follows:

$$\begin{aligned} \text{pattern number} &= 12388/64 = 192 \\ \text{POKE } 3948, 192 \end{aligned}$$

Rather than the character set, you may not use the area in bank 0 which is occupied by the character ROM image. It is also important to set the memory area in zero page ram. The usable block is from 0000 to 4095, for example, blocks 128 to 255.

The next step is to position your sprite. The useful screen area occupies a horizontal (X) position from 8000 to X=144, and a vertical position (Y) from Y=0 to Y=240. Each sprite has a register for its vertical position. Since the horizontal position can be larger than 255, two registers are used. Each sprite has a register for the low byte of its X position, and there's a single register for the high byte of the X position. The registers are found like:

$$\begin{aligned} \text{X register} &= 12288 + \text{Sprite No} \times 4 \\ \text{if you want registers 12288 + Sprite No} \times 4 \\ \text{if you want registers 12288} \end{aligned}$$

Before we move on, we must consider how a register holds sprite data. Each sprite has a specific bit which determines how an attribute is set. Sprites 0 use bit 4, sprite 5 use bit 5 and so on.

Using the above relationship, the following register will set up a sprite's position. X and Y are the sprite's position, and SN is the sprite's number:

$$\begin{aligned} \text{16 0000 0000000000000000} & \text{ X} \\ \text{00 00000000000000000000} & \\ \text{16 0000 0000000000000000} & \text{ Y} \\ \text{00 0000 0000 0000000000000000} & \text{ SN} \end{aligned}$$

Lines 50 and 60 show the map of tagging bits in sprite registers. Line 50 turns the bit on, and line 60 turns it off. This approach is used with the other registers. Table 1 summarizes the sprite registers for each you apply the following relationships:

$$\begin{aligned} \text{Base register} &= 1024 \times \text{Sprite} \times \text{Pattern} \times 1600 \\ \text{Base register} &= 1024 \times \text{Sprite} \times \text{Pattern} \times 1600 \end{aligned}$$

BIT PATTERNS	COLORS SOURCES
0000	WHITE
0001	Upper 4 bits of colour
0010	Upper 4 bits of colour from memory
0011	Lower 4 bits of colour
0100	Upper 4 bits of colour from memory
0101	Upper 4 bits of colour from memory
0110	Lower 4 bits of colour
0111	Lower 4 bits of colour

The priority register determines the position of the sprite relative to the screen contents. If the relevant bit is set, the sprite is behind the screen contents. If the bit is cleared, the sprite is in front of the screen contents. Sprites in sprite priorities are determined by the sprite number. Sprites 7 is highest and sprite 0 is lowest on the screen.

As with characters, you have the option of using single colour high resolution sprites or four colour high resolution sprites or four colour multicolour sprites. There are eight colour registers which specify the high resolution colour. They are found by using:

$$\text{Colour reg} = 53280 + 8N$$

Multicolour mode uses bit pairs, and the colours are derived in the following way:

BIT PATTERNS	COLORS SOURCES
00	WHITE
01	WHITE
10	Coloured register
11	WHITE

When using sprites, you may want to detect whether they collide with other screen contents. Two sprite collision registers are used.

Register 53276 detects collisions between sprites. The bits relevant to the sprite involved are set if there is a collision. If, for example, sprites 0 and 5 collide, then bits 0 and 5 are set. When you read this register to decide if a collision has occurred, the register is automatically cleared.

Register 53279 detects collisions between sprites and the characters on the screen. It functions in the same way as the sprite-to-sprite collision register.

With that's the theory. Next time, I'll give a set of assembly routines which will take some of the burden out of using graphics on the 64.

JiffyDOS

Disk Drive Replacement System

Jiffy Dos



Can the Jiffy defeat the dolphin?

By S. Garton

The combination of C64 and disk drive has often been described as a lumbering hippo, the reason for this being that disk access is extremely slow. A good indication of just how slow a C64 drive is, is that many games load quicker from cassette than from disk - silly, isn't it?

There are a variety of options available to disk users to help speed up disk access and loading time. These are:

Software fast loader Cartridge Replacement DOS

The first method of disk speeding is OK, when it works. But often, clashes between the memory used by a fast loader and the program you're trying to load prevent only a speed up system being used. Another problem is that a fast loader for one make of drive will not necessarily work on a different Commodore drive.

A cartridge-based fast loader usually gives a better chance of loading a program than the software method, since many cartridges take up little or no user RAM. However, there are problems. Firstly, you have to plug them in to your machine when you want to use them, and, despite warnings, many people have damaged their

computers by plugging in or unplugging cartridges while the machine is switched on.

Secondly, some cartridges require you to load fast load software into them, so by the time you've loaded the software in, you may as well have loaded the program in disk mode anyway. Still other cartridges require that you format, or copy, the program. Unfortunately, not all programs can be copied.

The third method of disk speeding is, in my mind, the best. A replacement DOS system usually requires you to open up both your disk drive and C64 or C128, remove some chips and add new ones containing the new disk access software. Two such systems are currently available. Firstly, there is Dolphin DOS, which has been around for some time. Now there is a new system on offer called Jiffy DOS.

I've been using Dolphin Dos for some time now, and am particularly happy with it. Dolphin DOS not only replaces chips inside the computer, but a new lead is used for communication between drive and computer. However, that lead plugs into the rear port at the rear of the computer, which is a real inconvenience at times, since I use this port for other things such as my modem.

Since I'm so pleased with Dolphin Dos, I was at first reluctant to try out the new replacement DOS system from Financed Systems Software - Jiffy DOS. However when I did, I was pleasantly surprised by what was on offer.

What You Get

One great problem with Dolphin DOS is that it will only work with my 1541 disk drive. I have a second drive, but this is an IIc extension, and I've been unable to find a fast DOS system to work with this. Until now that is - Jiffy DOS is available for all of the following computers and disk drives:

C64, 64C, 88-66, C128, C128D, 1541, 1541C, 1541-D, 1570, 1581, PS-D-1, PS-D-3, Eurocomstar, Euro 2001, as well as some others.

The fact that you've probably never heard of some of these disk drives will probably indicate to you that Jiffy DOS is of American origin. The originator of Jiffy DOS in the States are Creative Micro Designs, Inc. As you can see from the company list above, this replacement DOS is available for a much wider range of systems than any other system available.

The first thing that I noticed about Jiffy DOS was the lack of parts so it

—give me the completed boards and leads that I associate with my Dolphin DOS. If you have a C64, you simply get two chips with sockets attached. If you have a C128, you get three chips, two of them connected to a single socket. One of these chips is the replacement ROM to go inside your disk drive. The other chip (or chips), is a replacement ROM for the KERNAL chip (or chips), inside your computer.

Installation of the new chips is both quick and simple — all that is needed is a screwdriver and a little common sense. Clear, step-by-step installation instructions are included in the package.

What It Does

Jiffy DOS is designed to speed up all disk operations, including LOAD, SAVE, FORMAT, etc. The package claims increases of up to 15 times the speed of a standard drive. A DOS wedge is included in the new chips. This adds 84 new commands to Basic which give you much quicker access to your disks. Single key LOAD, SAVE, and directory commands are also added.

If you have a C128 then the DOS works in both C64 and C128 modes. Rather than spouting on about all of the commands that are offered by Jiffy Dos, I've listed all of the new commands in Figure 1.

If you want to see just how well Jiffy DOS performed, then you should check out Figure 2. Please note that figures are for a C128 with an Excelerator disk drive. Other combinations may give different results.

How Was It?

So what did I think of Jiffy DOS? The lack of a parallel cable was the first good point I noticed. I would have my drive where I wanted it, serial cable allowing cables than otherwise a short parallel lead prevented I have it. Secondly, the ability to use a smaller Exciterator drive rather than my bulky 1941 was of great benefit.

On the software side, I found that most programs that I used worked without any problems, though some heavily processed software would not load without me disabling the new DOS.

Disabling the DOS simply required that the switch on the board be turned off. Jiffy DOS can be switched on and

run while the computer is still on. This has the advantage that should a program not load while the DOS was turned on, I could simply switch it off. Once the program had loaded, I could turn the DOS back on as needed to speed up any further disk activity.

Dolphin DOS presents a C128 owner with a slight problem: if the C128 Kernal is replaced, then you lose the ability to load software from cassette tape. If you should ever need to use tape in C128 mode, you'd have to open up the machine and put your old Kernal back in. Plugging the Jiffy DOS switch turns the DOS completely off, and tape use is then allowed, a great benefit for C128 users who require tape.

Of course Jiffy DOS isn't all a bed of roses, and there are a number of areas where I feel Dolphin DOS has the edge on it. For starters I did run being able to access the monitor that's built into Dolphin DOS. The machine code monitor that Dolphin DOS

provides may only be a simple one, but it does come in handy at times.

Dolphin DOS is quicker than Jiffy DOS, and having used Dolphin DOS for some time, I did sometimes find myself waiting for something to load. The cheaper price of Jiffy DOS and its ability to work with just about any system configuration does make it an excellent product. Dolphin DOS offers a few more features, but does limit you to the type of disk drive you can use.

Being a C128 user who quite often needs to use a cassette recorder as 128 mode, turn the user part for plugging a number of items into the computer and requires quick disk access, Jiffy DOS does everything that I require, and would be my choice over Dolphin DOS, even if it isn't quite as fast.

Finalizer

Product: Jiffy Dos **Supplier:** FISH-L, 18 High Street, Farnham, Surrey, GU10 1RG. Tel: (01825) 553155. **Price:** £14.95

Command Summary

Standard DOS 5.1 Wedge Commands

0	Read the disk drive control channel
0C: newFile=File	Copy a file to the name specified
0D:	Format the disk drive
0E: directory	Format (DIR) the disk drive
0F: directory	Open DIR
10:	Display the Jiffy DOS commands
11: command=command	Prints a file
12: file1[, file2]	Search a file for file2
13:	Mount the disk drive
14:	Unmount a disk
15:	Display the disk drive error
16: file	See the disk drive, number
17: file	Load a BASIC program
18: file	Load and run a BASIC program
19: file	Load an ML program
1A: file	Run a BASIC program

Additional JiffyDOS Commands

0B:	Display/modify the 1941 head state
1C: file	Load a BASIC program from disk
1D:	Display/modify the format type
1E: file	Load/modify a file
1F: file	Load an ASCII file from disk
20:	Load a BASIC program
21: file	Load an ML file from disk
22: file	Load an ML file from disk
23: file	Load an ML file from disk
24: file	Load an ML file from disk
25: file	Load an ML file from disk
26: file	Load an ML file from disk
27: file	Load an ML file from disk
28: file	Load an ML file from disk
29: file	Load an ML file from disk
2A: file	Load an ML file from disk
2B: file	Load an ML file from disk
2C: file	Load an ML file from disk
2D: file	Load an ML file from disk
2E: file	Load an ML file from disk
2F: file	Load an ML file from disk
30: file	Load an ML file from disk
31: file	Load an ML file from disk
32: file	Load an ML file from disk
33: file	Load an ML file from disk
34: file	Load an ML file from disk
35: file	Load an ML file from disk
36: file	Load an ML file from disk
37: file	Load an ML file from disk
38: file	Load an ML file from disk
39: file	Load an ML file from disk
3A: file	Load an ML file from disk
3B: file	Load an ML file from disk
3C: file	Load an ML file from disk
3D: file	Load an ML file from disk
3E: file	Load an ML file from disk
3F: file	Load an ML file from disk
40: file	Load an ML file from disk
41: file	Load an ML file from disk
42: file	Load an ML file from disk
43: file	Load an ML file from disk
44: file	Load an ML file from disk
45: file	Load an ML file from disk
46: file	Load an ML file from disk
47: file	Load an ML file from disk
48: file	Load an ML file from disk
49: file	Load an ML file from disk
4A: file	Load an ML file from disk
4B: file	Load an ML file from disk
4C: file	Load an ML file from disk
4D: file	Load an ML file from disk
4E: file	Load an ML file from disk
4F: file	Load an ML file from disk
50: file	Load an ML file from disk
51: file	Load an ML file from disk
52: file	Load an ML file from disk
53: file	Load an ML file from disk
54: file	Load an ML file from disk
55: file	Load an ML file from disk
56: file	Load an ML file from disk
57: file	Load an ML file from disk
58: file	Load an ML file from disk
59: file	Load an ML file from disk
5A: file	Load an ML file from disk
5B: file	Load an ML file from disk
5C: file	Load an ML file from disk
5D: file	Load an ML file from disk
5E: file	Load an ML file from disk
5F: file	Load an ML file from disk
60: file	Load an ML file from disk
61: file	Load an ML file from disk
62: file	Load an ML file from disk
63: file	Load an ML file from disk
64: file	Load an ML file from disk
65: file	Load an ML file from disk
66: file	Load an ML file from disk
67: file	Load an ML file from disk
68: file	Load an ML file from disk
69: file	Load an ML file from disk
6A: file	Load an ML file from disk
6B: file	Load an ML file from disk
6C: file	Load an ML file from disk
6D: file	Load an ML file from disk
6E: file	Load an ML file from disk
6F: file	Load an ML file from disk
70: file	Load an ML file from disk
71: file	Load an ML file from disk
72: file	Load an ML file from disk
73: file	Load an ML file from disk
74: file	Load an ML file from disk
75: file	Load an ML file from disk
76: file	Load an ML file from disk
77: file	Load an ML file from disk
78: file	Load an ML file from disk
79: file	Load an ML file from disk
7A: file	Load an ML file from disk
7B: file	Load an ML file from disk
7C: file	Load an ML file from disk
7D: file	Load an ML file from disk
7E: file	Load an ML file from disk
7F: file	Load an ML file from disk
80: file	Load an ML file from disk
81: file	Load an ML file from disk
82: file	Load an ML file from disk
83: file	Load an ML file from disk
84: file	Load an ML file from disk
85: file	Load an ML file from disk
86: file	Load an ML file from disk
87: file	Load an ML file from disk
88: file	Load an ML file from disk
89: file	Load an ML file from disk
8A: file	Load an ML file from disk
8B: file	Load an ML file from disk
8C: file	Load an ML file from disk
8D: file	Load an ML file from disk
8E: file	Load an ML file from disk
8F: file	Load an ML file from disk
90: file	Load an ML file from disk
91: file	Load an ML file from disk
92: file	Load an ML file from disk
93: file	Load an ML file from disk
94: file	Load an ML file from disk
95: file	Load an ML file from disk
96: file	Load an ML file from disk
97: file	Load an ML file from disk
98: file	Load an ML file from disk
99: file	Load an ML file from disk
9A: file	Load an ML file from disk
9B: file	Load an ML file from disk
9C: file	Load an ML file from disk
9D: file	Load an ML file from disk
9E: file	Load an ML file from disk
9F: file	Load an ML file from disk
100: file	Load an ML file from disk
101: file	Load an ML file from disk
102: file	Load an ML file from disk
103: file	Load an ML file from disk
104: file	Load an ML file from disk
105: file	Load an ML file from disk
106: file	Load an ML file from disk
107: file	Load an ML file from disk
108: file	Load an ML file from disk
109: file	Load an ML file from disk
10A: file	Load an ML file from disk
10B: file	Load an ML file from disk
10C: file	Load an ML file from disk
10D: file	Load an ML file from disk
10E: file	Load an ML file from disk
10F: file	Load an ML file from disk
110: file	Load an ML file from disk
111: file	Load an ML file from disk
112: file	Load an ML file from disk
113: file	Load an ML file from disk
114: file	Load an ML file from disk
115: file	Load an ML file from disk
116: file	Load an ML file from disk
117: file	Load an ML file from disk
118: file	Load an ML file from disk
119: file	Load an ML file from disk
11A: file	Load an ML file from disk
11B: file	Load an ML file from disk
11C: file	Load an ML file from disk
11D: file	Load an ML file from disk
11E: file	Load an ML file from disk
11F: file	Load an ML file from disk
120: file	Load an ML file from disk
121: file	Load an ML file from disk
122: file	Load an ML file from disk
123: file	Load an ML file from disk
124: file	Load an ML file from disk
125: file	Load an ML file from disk
126: file	Load an ML file from disk
127: file	Load an ML file from disk
128: file	Load an ML file from disk
129: file	Load an ML file from disk
12A: file	Load an ML file from disk
12B: file	Load an ML file from disk
12C: file	Load an ML file from disk
12D: file	Load an ML file from disk
12E: file	Load an ML file from disk
12F: file	Load an ML file from disk
130: file	Load an ML file from disk
131: file	Load an ML file from disk
132: file	Load an ML file from disk
133: file	Load an ML file from disk
134: file	Load an ML file from disk
135: file	Load an ML file from disk
136: file	Load an ML file from disk
137: file	Load an ML file from disk
138: file	Load an ML file from disk
139: file	Load an ML file from disk
13A: file	Load an ML file from disk
13B: file	Load an ML file from disk
13C: file	Load an ML file from disk
13D: file	Load an ML file from disk
13E: file	Load an ML file from disk
13F: file	Load an ML file from disk
140: file	Load an ML file from disk
141: file	Load an ML file from disk
142: file	Load an ML file from disk
143: file	Load an ML file from disk
144: file	Load an ML file from disk
145: file	Load an ML file from disk
146: file	Load an ML file from disk
147: file	Load an ML file from disk
148: file	Load an ML file from disk
149: file	Load an ML file from disk
14A: file	Load an ML file from disk
14B: file	Load an ML file from disk
14C: file	Load an ML file from disk
14D: file	Load an ML file from disk
14E: file	Load an ML file from disk
14F: file	Load an ML file from disk
150: file	Load an ML file from disk
151: file	Load an ML file from disk
152: file	Load an ML file from disk
153: file	Load an ML file from disk
154: file	Load an ML file from disk
155: file	Load an ML file from disk
156: file	Load an ML file from disk
157: file	Load an ML file from disk
158: file	Load an ML file from disk
159: file	Load an ML file from disk
15A: file	Load an ML file from disk
15B: file	Load an ML file from disk
15C: file	Load an ML file from disk
15D: file	Load an ML file from disk
15E: file	Load an ML file from disk
15F: file	Load an ML file from disk
160: file	Load an ML file from disk
161: file	Load an ML file from disk
162: file	Load an ML file from disk
163: file	Load an ML file from disk
164: file	Load an ML file from disk
165: file	Load an ML file from disk
166: file	Load an ML file from disk
167: file	Load an ML file from disk
168: file	Load an ML file from disk
169: file	Load an ML file from disk
16A: file	Load an ML file from disk
16B: file	Load an ML file from disk
16C: file	Load an ML file from disk
16D: file	Load an ML file from disk
16E: file	Load an ML file from disk
16F: file	Load an ML file from disk
170: file	Load an ML file from disk
171: file	Load an ML file from disk
172: file	Load an ML file from disk
173: file	Load an ML file from disk
174: file	Load an ML file from disk
175: file	Load an ML file from disk
176: file	Load an ML file from disk
177: file	Load an ML file from disk
178: file	Load an ML file from disk
179: file	Load an ML file from disk
17A: file	Load an ML file from disk
17B: file	Load an ML file from disk
17C: file	Load an ML file from disk
17D: file	Load an ML file from disk
17E: file	Load an ML file from disk
17F: file	Load an ML file from disk
180: file	Load an ML file from disk
181: file	Load an ML file from disk
182: file	Load an ML file from disk
183: file	Load an ML file from disk
184: file	Load an ML file from disk
185: file	Load an ML file from disk
186: file	Load an ML file from disk
187: file	Load an ML file from disk
188: file	Load an ML file from disk
189: file	Load an ML file from disk
18A: file	Load an ML file from disk
18B: file	Load an ML file from disk
18C: file	Load an ML file from disk
18D: file	Load an ML file from disk
18E: file	Load an ML file from disk
18F: file	Load an ML file from disk
190: file	Load an ML file from disk
191: file	Load an ML file from disk
192: file	Load an ML file from disk
193: file	Load an ML file from disk
194: file	Load an ML file from disk
195: file	Load an ML file from disk
196: file	Load an ML file from disk
197: file	Load an ML file from disk
198: file	Load an ML file from disk
199: file	Load an ML file from disk
19A: file	Load an ML file from disk
19B: file	Load an ML file from disk
19C: file	Load an ML file from disk
19D: file	Load an ML file from disk
19E: file	Load an ML file from disk
19F: file	Load an ML file from disk
200: file	Load an ML file from disk
201: file	Load an ML file from disk
202: file	Load an ML file from disk
203: file	Load an ML file from disk
204: file	Load an ML file from disk
205: file	Load an ML file from disk
206: file	Load an ML file from disk
207: file	Load an ML file from disk
208: file	Load an ML file from disk
209: file	Load an ML file from disk
20A: file	Load an ML file from disk
20B: file	Load an ML file from disk
20C: file	Load an ML file from disk
20D: file	Load an ML file from disk
20E: file	Load an ML file from disk
20F: file	Load an ML file from disk
210: file	Load an ML file from disk
211: file	Load an ML file from disk
212: file	Load an ML file from disk
213: file	Load an ML file from disk
214: file	Load an ML file from disk
215: file	Load an ML file from disk
216: file	Load an ML file from disk
217: file	Load an ML file from disk
218: file	Load an ML file from disk
219: file	Load an ML file from disk
21A: file	Load an ML file from disk
21B: file	Load an ML file from disk
21C: file	Load an ML file from disk
21D: file	Load an ML file from disk
21E: file	Load an ML file from disk
21F: file	Load an ML file from disk
220: file	Load an ML file from disk
221: file	Load an ML file from disk
222: file	Load an ML file from disk
223: file	Load an ML file from disk
224: file	Load an ML file from disk
225: file	Load an ML file from disk
226: file	Load an ML file from disk
227: file	Load an ML file from disk
228: file	Load an ML file from disk
229: file	Load an ML file from disk
22A: file	Load an ML file from disk
22B: file	Load an ML file from disk
22C: file	Load an ML file from disk
22D: file	Load an ML file from disk
22E: file	Load an ML file from disk
22F: file	Load an ML file from disk
230: file	Load an ML file from disk
231: file	Load an ML file from disk
232: file	Load an ML file from disk
233: file	Load an ML file from disk
234: file	Load an ML file from disk
235: file	Load an ML file from disk
236: file	Load an ML file from disk
237: file	Load an ML file from disk
238: file	Load an ML file from disk
239: file	Load an ML file from disk
23A: file	Load an ML file from disk
23B: file	Load an ML file from disk
23C: file	Load an ML file from disk
23D: file	Load an ML file from disk
23E: file	Load an ML file from disk
23F: file	Load an ML file from disk
240: file	Load an ML file from disk
241: file	Load an ML file from disk
242: file	Load an ML file from disk
243: file	Load an ML file from disk
244: file	Load an ML file from disk
245: file	Load an ML file from disk
246: file	Load an ML file from disk
247: file	Load an ML file from disk
248: file	Load an ML file from disk
249: file	Load an ML file from disk
24A: file	Load an ML file from disk
24B: file	Load an ML file from disk
24C: file	Load an ML file from disk
24D: file	Load an ML file from disk
24E: file	Load an ML file from disk
24F: file	Load an ML file from disk
250: file	Load an ML file from disk
251: file	Load an ML file from disk
252: file	Load an ML file from disk
253: file	Load an ML file from disk
254: file	Load an ML file from disk
255: file	Load an ML file from disk
256: file	Load an ML file from disk
257: file	Load an ML file from disk
258: file	Load an ML file from disk

Mouse 80

Do you own a C128 and a mouse? Then here's the utility to use them together in 80-column text mode

By D.H. Faber

When I recently purchased a mouse (Commodore 1351), I was not too surprised to find that the accompanying software contained a C128 mouse driver for the 80-column mode only. After all, the 8563 VDC chip (or 1566 in the newer models with 64K video ram) does not support sprites, and as complicated screen-matched doesn't help in creating software sprites either!

However, I intended to use my new acquisition mainly as my own program for the selection of options (or features, etc.) from the 80-column screen. If you have similar needs, the utility **MOUSE80** described below is the answer. If you don't own a mouse, please read on anyway, because pay-text method is also provided!

How it works

At this point, let's assume you've already spent some time typing in the programs **EDIT** ("editing it"), and you have the machine code utility **MOUSE80** and the basic demo program **DemoMOUSE80-BAS** at hand. Let's combine an X-ray of the utility's inner life with some action on your part! Connect the mouse to either port (if no mouse signals are detected on port 1, then port 2 is used), or, if you don't have a mouse plug in a joystick (instead if no mouse signals are detected on either port, then joystick control is assumed), note that these tests occur within the **MOUSE80** utility itself, not in the demo program.

Now load and run **DemoMOUSE80-BAS** (type enter before the first thing it wants to do to load **MOUSE80**). If all is well, you'll see a screen with lines numbered from 0 to 21, using various colours for each line (if you like us - have to make do with a monochrome half-tone you'll see an additional half-tone only).

On the upper half of the screen the two character sets are displayed, the lower half shows a cursor option that can be selected. The mouse's cursor takes the shape of an arrow,

usually located somewhere in the lower region of the screen (**MOUSE80** was written for the Commodore 1351 mouse, since I cannot vouch for the compatibility of other brands, you'll have to try them for yourself).

Feel free to move the arrow around, but don't press any buttons yet! Anyhow, even will note that the arrow's acceleration as the movement continues in speed is doubled with each of the four available "pages", with a proportional mouse there's no need for such acceleration.

Notice anything funny on the fourth line? There are six adjacent characters showing fragments of the arrow and the characters it is currently moving over. This has to do with the way the left-point is generated: the cursor (arrow) affects up to six character positions, the last pattern is taken from the character set, the sprite is overlaid and the six "spurred" characters are used to display the "spurred" ones.

This process is repeated continuously until one of the mouse buttons (or the last button of the joystick) is pressed. The six characters (used for this purpose are rather obscure ones:

REVERSE-CRM-Y/U/O/H/I/L (hexadecimal display codes 244-249 in the graphic set).

This is no great loss, especially since the other characters are still available in the alternative set. Anyway, if you don't like the above you can select a different set of six adjacent characters to be used for this purpose (see below).

Two more things are noticeable about the arrow's movement. Firstly, it doesn't leave the screen area. In the vicinity of the borders it even changes direction to be able to point at the borders "from within". Secondly, although it moves point-by-point in a vertical direction, in horizontal direction it moves two pixels at a time.

This was done not out of necessity, but for practical reasons only: it

reduces the required misalignment, which otherwise would be twice as large horizontally as for the 40-column screen. Also, this utility was not designed for graphical packages, so greater accuracy would be superfluous.

Now, regarding the "information area" for the time being: any "clicking" on various places on the screen. In the right-hand bottom corner of the screen, you may read at what position exactly you clicked. Here the character positions are counted from zero onwards (in a Basic's **CHAR** instruction). Note that the maximum position in horizontal direction is 628 instead of 639, since the arrow points to the bottom point of a pair of pixels (see above).

What happens if, after pressing a button, nothing is returned to the calling program, which can then assume the latest cursor position and decide what to do next (if not on an **OPTION** area, continue with the arrow in the same position). The calling program can also detect which button (in case a mouse is used) was pressed (see below). This demo program treats both buttons alike.

You may have noticed that the arrow takes on the colour of the character it is displayed on, unlike the normal cursor which temporarily "points" the character it is on. There exists another possibility however: try clicking on **MONOCHROME**. This results in a screen in one colour only, but some characters are now looking weird.

What's happened? If the **ATN** line is active (IS of the VDC is turned off), then the foreground colour is taken from the bit 0 of register 35 and not from the attribute. However, the processor doesn't look at the attribute for the character but either? Instead, for all characters the graphic set is assumed. Therefore, if you want to use the monochrome mode, build your options screen with characters from the graphic set only (you may consider swapping the sets in video ram to use characters from the other set instead).

The most remark on this the colour used for the monochrome

option is the one used in the most recent PRINT statement as stored in location 240 (*SF0). You may also enter the desired colour code directly into the location.

To continue our guided tour, click on COLOR to return the multi-coloured screen. Up to now we've used a steady arrow. If you find it's difficult at times to locate its position, especially on a crowded part of the screen, it's clicking on BLINK FASTER and BLINK SLOWER. You'll find that you may choose between a steady cursor (speed=0) and one that blinks slowly (speed=128) or quickly (speed=4). This option may also be easily initiated from the calling program when using MOUSE568 (see below).

Before ending the demo by clicking QUIT, some more remarks are in order. The VIC must be in 40-column mode, 256 pixels per character (40-line power-up). You may change the start addresses of screen, attributes and character sets, and you may also change the number of lines displayed. If you're not sure how to do this, it may help to consult the listing of the demo program.

For the sake of completeness, MOUSE568 is not wedged into the IRQ routine as it usually done with mouse demos. The reason is to be found in the VIC's complicated screen method. Changing its address (the contents of a location is taken out for over a register) requires a multitude of machine code instructions, and another program trying to access the VIC can easily corrupt some of its contents, with unpredictable results. Therefore MOUSE568 retains control only after a mouse button has been pressed in the meantime a very robust interrupt to avoid keyboard interference.

In use

The machine code file MOUSE568 loads from \$E000 to \$E9F0 in RAM 0 (the area from \$E9F0 to \$E9FF is used as a scratch pad). From BASIC the mouse driver is called as

```
BANK0 SYS$TM4(A,X,Y)
```

The meaning of the parameters A, X and Y (which are transferred to the same X-registers and Y-registers respectively) is as follows:

X = horizontal character position at

which the sprite will first appear (0-255).

Y = slow vertical character position. Normal range 0-24 or higher if you changed the number of lines displayed.

Note that the parameters are not checked! A special case is X=Y=255: the sprite will reappear at the same spot it was in when the previous call to MOUSE568 returned control to the calling program. This is useful to restart the driver if you click on an irrelevant part of the screen.

A = blink speed 0 (for no blink, 1 to 12 for fast to slow blink. If A < 16 the mouse driver operates in colour mode. To use monochrome mode add 128 to the value of A, the colour used is taken from the most recent PRINT statement or may be pointed directly to location 240 (*SF0).

Control is returned to the calling program after pressing one of the

mouse buttons (or the Exit button of a joystick). Relevant information is stored in registers A, X and Y. To obtain these values use

```
BANK0 A,X,Y
```

(A) is zero if the left mouse button was pressed, 1 for the right button (its value is irrelevant for a joystick). The values of X and Y are the horizontal and vertical character positions respectively at which the arrow was pointing. You may find the exact pixel values as follows:

```
BANK0
SYS$M42(BUREG_A,X)PIX = A + 256*X
SYS$M42(BUREG_A,X,Y)PIX = A + 256*X
```

If you want to modify different characters to generate the soft sprite,



you should

POKE the screen display code of the first one to location \$D905 (15483D), the default value is 244.
POKE the character(s) 0 or 1 to location \$D90F (15487F), default is 0 (the graphics set).

For ASSEMBLER programs you may call the driver from anywhere in RAM0 below \$D900.

```
LDA #13F
STA $DFA0
Out A, X, Y (as above)
JSR $D900
```

On return, A, X and Y contain values whose meaning is as described above. To obtain the exact pixel location of the screen set

```
LDA #13F
STA $DFA0
JSR $D90A
```

to obtain the low- and high byte of the x-value in A, and X respectively. Use JSR \$D90B likewise to obtain the y-value.

If you wish to call these routines from underneath the I/O area (\$D900-\$D90F), or from a different RAM bank or a cartridge or the famous ROM, you'll have to use the formal JSRPAR routine. If you really intend to do this, I suspect you'll know how to write, if not consult a thoroughly documented ROM listing.

MOUSEPAD of course contains instructions to access the VDC's registers and the video ram. As a bonus

to assembly programmers, here's how to use them for your own purposes.

Assuming the calling program is in RAM0 and not underneath the I/O area, you must select a bank with RAM0 and the I/O components.

```
LDA #53E
STA $DFA0
```

direct \$D900-\$D90F or from other banks, you must use JSRPAR, not above. You can now use live subroutines as follows:

```
REGWRITE JSR $D905 (A,X)
value X is stored in register A
REGREAD JSR $D90F (A) - value
of register A is stored in A
VDCWRITE JSR $D908 (A,X,Y) -
value Y is stored in video ram at A /
X (low/high byte)
VDCREAD JSR $D903 (A,X) - the
value in video ram (A/X = low/high
byte) is transferred to A
SETUPDATE JSR $D90D (A,X)
the contents of A and X are transferred
to registers 19 and 18 (UPDATE low
and high byte respectively)
```

To Bank programmers, these subroutines are not available since there exists no BANK, assumed to select the required memory configuration. However, if you consult the listing of the demo program and copy the DATA statements and the boot-poking there into memory, you'll have the same facilities available in BANK15 by

```
REGWRITE SYS252A X
REGREAD SYS264 A
```

```
VDCWRITE SYS209A,X,Y
VDCREAD SYS248A,X
SETUPDATE SYS210A,X
```

Getting it in

In the listings you'll find the two Bank loaders: INTER64/DEMO-BAS and INTER4/INTER-Basic and save these using VCR system checker (in 64 the following pages)

mode/). Next (only in 64 mode) enter POK\$45, 51 POK\$44,41 POK\$45370,0 and LOAD "INTER64/DEMO-BAS". (For 1 for tape units, there should also change 5 to 1 in line 120 of the program). Now run it, the Basic program DEMONSTRATES BAS is written to disk (or tape), ready to be used in 128 mode.

Next type "NEW" and (again in 64 mode) enter the same pokes to above and LOAD "INTER4/INTER-BASIC" (note: you should change the 5 to 1, also in line 110 of the program) and run it. It saves INTER4 to disk or tape. Now start your computer in 128 mode, load INTER4 (tape unit change 5 to 1 in line 88) and run it. This produces the machine code file MOUSE68.

One final note for tape users: remove the BLOAD from line 14 of the demo program and LOAD "MOUSE 89",1 in direct mode before loading and running the demo program!



PROGRAM INTER4/DEMO-BAS

```
10 18 REM --- INTER4/DEMO-BAS ---
20 20 IF $DFA0 = 0 THEN GOTO 30
30 20 IF $DFA0 = 1 THEN GOTO 40
40 20 IF $DFA0 = 2 THEN GOTO 50
50 20 IF $DFA0 = 3 THEN GOTO 60
60 20 IF $DFA0 = 4 THEN GOTO 70
70 20 IF $DFA0 = 5 THEN GOTO 80
80 20 IF $DFA0 = 6 THEN GOTO 90
90 20 IF $DFA0 = 7 THEN GOTO 100
100 20 IF $DFA0 = 8 THEN GOTO 110
110 20 IF $DFA0 = 9 THEN GOTO 120
120 20 IF $DFA0 = 10 THEN GOTO 130
130 20 IF $DFA0 = 11 THEN GOTO 140
140 20 IF $DFA0 = 12 THEN GOTO 150
150 20 IF $DFA0 = 13 THEN GOTO 160
160 20 IF $DFA0 = 14 THEN GOTO 170
170 20 IF $DFA0 = 15 THEN GOTO 180
180 20 IF $DFA0 = 16 THEN GOTO 190
190 20 IF $DFA0 = 17 THEN GOTO 200
200 20 IF $DFA0 = 18 THEN GOTO 210
210 20 IF $DFA0 = 19 THEN GOTO 220
220 20 IF $DFA0 = 20 THEN GOTO 230
230 20 IF $DFA0 = 21 THEN GOTO 240
240 20 IF $DFA0 = 22 THEN GOTO 250
250 20 IF $DFA0 = 23 THEN GOTO 260
260 20 IF $DFA0 = 24 THEN GOTO 270
270 20 IF $DFA0 = 25 THEN GOTO 280
280 20 IF $DFA0 = 26 THEN GOTO 290
290 20 IF $DFA0 = 27 THEN GOTO 300
300 20 IF $DFA0 = 28 THEN GOTO 310
310 20 IF $DFA0 = 29 THEN GOTO 320
320 20 IF $DFA0 = 30 THEN GOTO 330
330 20 IF $DFA0 = 31 THEN GOTO 340
340 20 IF $DFA0 = 32 THEN GOTO 350
350 20 IF $DFA0 = 33 THEN GOTO 360
360 20 IF $DFA0 = 34 THEN GOTO 370
370 20 IF $DFA0 = 35 THEN GOTO 380
380 20 IF $DFA0 = 36 THEN GOTO 390
390 20 IF $DFA0 = 37 THEN GOTO 400
400 20 IF $DFA0 = 38 THEN GOTO 410
410 20 IF $DFA0 = 39 THEN GOTO 420
420 20 IF $DFA0 = 40 THEN GOTO 430
430 20 IF $DFA0 = 41 THEN GOTO 440
440 20 IF $DFA0 = 42 THEN GOTO 450
450 20 IF $DFA0 = 43 THEN GOTO 460
460 20 IF $DFA0 = 44 THEN GOTO 470
470 20 IF $DFA0 = 45 THEN GOTO 480
480 20 IF $DFA0 = 46 THEN GOTO 490
490 20 IF $DFA0 = 47 THEN GOTO 500
500 20 IF $DFA0 = 48 THEN GOTO 510
510 20 IF $DFA0 = 49 THEN GOTO 520
520 20 IF $DFA0 = 50 THEN GOTO 530
530 20 IF $DFA0 = 51 THEN GOTO 540
540 20 IF $DFA0 = 52 THEN GOTO 550
550 20 IF $DFA0 = 53 THEN GOTO 560
560 20 IF $DFA0 = 54 THEN GOTO 570
570 20 IF $DFA0 = 55 THEN GOTO 580
580 20 IF $DFA0 = 56 THEN GOTO 590
590 20 IF $DFA0 = 57 THEN GOTO 600
600 20 IF $DFA0 = 58 THEN GOTO 610
610 20 IF $DFA0 = 59 THEN GOTO 620
620 20 IF $DFA0 = 60 THEN GOTO 630
630 20 IF $DFA0 = 61 THEN GOTO 640
640 20 IF $DFA0 = 62 THEN GOTO 650
650 20 IF $DFA0 = 63 THEN GOTO 660
660 20 IF $DFA0 = 64 THEN GOTO 670
670 20 IF $DFA0 = 65 THEN GOTO 680
680 20 IF $DFA0 = 66 THEN GOTO 690
690 20 IF $DFA0 = 67 THEN GOTO 700
700 20 IF $DFA0 = 68 THEN GOTO 710
710 20 IF $DFA0 = 69 THEN GOTO 720
720 20 IF $DFA0 = 70 THEN GOTO 730
730 20 IF $DFA0 = 71 THEN GOTO 740
740 20 IF $DFA0 = 72 THEN GOTO 750
750 20 IF $DFA0 = 73 THEN GOTO 760
760 20 IF $DFA0 = 74 THEN GOTO 770
770 20 IF $DFA0 = 75 THEN GOTO 780
780 20 IF $DFA0 = 76 THEN GOTO 790
790 20 IF $DFA0 = 77 THEN GOTO 800
800 20 IF $DFA0 = 78 THEN GOTO 810
810 20 IF $DFA0 = 79 THEN GOTO 820
820 20 IF $DFA0 = 80 THEN GOTO 830
830 20 IF $DFA0 = 81 THEN GOTO 840
840 20 IF $DFA0 = 82 THEN GOTO 850
850 20 IF $DFA0 = 83 THEN GOTO 860
860 20 IF $DFA0 = 84 THEN GOTO 870
870 20 IF $DFA0 = 85 THEN GOTO 880
880 20 IF $DFA0 = 86 THEN GOTO 890
890 20 IF $DFA0 = 87 THEN GOTO 900
900 20 IF $DFA0 = 88 THEN GOTO 910
910 20 IF $DFA0 = 89 THEN GOTO 920
920 20 IF $DFA0 = 90 THEN GOTO 930
930 20 IF $DFA0 = 91 THEN GOTO 940
940 20 IF $DFA0 = 92 THEN GOTO 950
950 20 IF $DFA0 = 93 THEN GOTO 960
960 20 IF $DFA0 = 94 THEN GOTO 970
970 20 IF $DFA0 = 95 THEN GOTO 980
980 20 IF $DFA0 = 96 THEN GOTO 990
990 20 IF $DFA0 = 97 THEN GOTO 1000
1000 20 IF $DFA0 = 98 THEN GOTO 1010
1010 20 IF $DFA0 = 99 THEN GOTO 1020
1020 20 IF $DFA0 = 100 THEN GOTO 1030
1030 20 IF $DFA0 = 101 THEN GOTO 1040
1040 20 IF $DFA0 = 102 THEN GOTO 1050
1050 20 IF $DFA0 = 103 THEN GOTO 1060
1060 20 IF $DFA0 = 104 THEN GOTO 1070
1070 20 IF $DFA0 = 105 THEN GOTO 1080
1080 20 IF $DFA0 = 106 THEN GOTO 1090
1090 20 IF $DFA0 = 107 THEN GOTO 1100
1100 20 IF $DFA0 = 108 THEN GOTO 1110
1110 20 IF $DFA0 = 109 THEN GOTO 1120
1120 20 IF $DFA0 = 110 THEN GOTO 1130
1130 20 IF $DFA0 = 111 THEN GOTO 1140
1140 20 IF $DFA0 = 112 THEN GOTO 1150
1150 20 IF $DFA0 = 113 THEN GOTO 1160
1160 20 IF $DFA0 = 114 THEN GOTO 1170
1170 20 IF $DFA0 = 115 THEN GOTO 1180
1180 20 IF $DFA0 = 116 THEN GOTO 1190
1190 20 IF $DFA0 = 117 THEN GOTO 1200
1200 20 IF $DFA0 = 118 THEN GOTO 1210
1210 20 IF $DFA0 = 119 THEN GOTO 1220
1220 20 IF $DFA0 = 120 THEN GOTO 1230
1230 20 IF $DFA0 = 121 THEN GOTO 1240
1240 20 IF $DFA0 = 122 THEN GOTO 1250
1250 20 IF $DFA0 = 123 THEN GOTO 1260
1260 20 IF $DFA0 = 124 THEN GOTO 1270
1270 20 IF $DFA0 = 125 THEN GOTO 1280
1280 20 IF $DFA0 = 126 THEN GOTO 1290
1290 20 IF $DFA0 = 127 THEN GOTO 1300
1300 20 IF $DFA0 = 128 THEN GOTO 1310
1310 20 IF $DFA0 = 129 THEN GOTO 1320
1320 20 IF $DFA0 = 130 THEN GOTO 1330
1330 20 IF $DFA0 = 131 THEN GOTO 1340
1340 20 IF $DFA0 = 132 THEN GOTO 1350
1350 20 IF $DFA0 = 133 THEN GOTO 1360
1360 20 IF $DFA0 = 134 THEN GOTO 1370
1370 20 IF $DFA0 = 135 THEN GOTO 1380
1380 20 IF $DFA0 = 136 THEN GOTO 1390
1390 20 IF $DFA0 = 137 THEN GOTO 1400
1400 20 IF $DFA0 = 138 THEN GOTO 1410
1410 20 IF $DFA0 = 139 THEN GOTO 1420
1420 20 IF $DFA0 = 140 THEN GOTO 1430
1430 20 IF $DFA0 = 141 THEN GOTO 1440
1440 20 IF $DFA0 = 142 THEN GOTO 1450
1450 20 IF $DFA0 = 143 THEN GOTO 1460
1460 20 IF $DFA0 = 144 THEN GOTO 1470
1470 20 IF $DFA0 = 145 THEN GOTO 1480
1480 20 IF $DFA0 = 146 THEN GOTO 1490
1490 20 IF $DFA0 = 147 THEN GOTO 1500
1500 20 IF $DFA0 = 148 THEN GOTO 1510
1510 20 IF $DFA0 = 149 THEN GOTO 1520
1520 20 IF $DFA0 = 150 THEN GOTO 1530
1530 20 IF $DFA0 = 151 THEN GOTO 1540
1540 20 IF $DFA0 = 152 THEN GOTO 1550
1550 20 IF $DFA0 = 153 THEN GOTO 1560
1560 20 IF $DFA0 = 154 THEN GOTO 1570
1570 20 IF $DFA0 = 155 THEN GOTO 1580
1580 20 IF $DFA0 = 156 THEN GOTO 1590
1590 20 IF $DFA0 = 157 THEN GOTO 1600
1600 20 IF $DFA0 = 158 THEN GOTO 1610
1610 20 IF $DFA0 = 159 THEN GOTO 1620
1620 20 IF $DFA0 = 160 THEN GOTO 1630
1630 20 IF $DFA0 = 161 THEN GOTO 1640
1640 20 IF $DFA0 = 162 THEN GOTO 1650
1650 20 IF $DFA0 = 163 THEN GOTO 1660
1660 20 IF $DFA0 = 164 THEN GOTO 1670
1670 20 IF $DFA0 = 165 THEN GOTO 1680
1680 20 IF $DFA0 = 166 THEN GOTO 1690
1690 20 IF $DFA0 = 167 THEN GOTO 1700
1700 20 IF $DFA0 = 168 THEN GOTO 1710
1710 20 IF $DFA0 = 169 THEN GOTO 1720
1720 20 IF $DFA0 = 170 THEN GOTO 1730
1730 20 IF $DFA0 = 171 THEN GOTO 1740
1740 20 IF $DFA0 = 172 THEN GOTO 1750
1750 20 IF $DFA0 = 173 THEN GOTO 1760
1760 20 IF $DFA0 = 174 THEN GOTO 1770
1770 20 IF $DFA0 = 175 THEN GOTO 1780
1780 20 IF $DFA0 = 176 THEN GOTO 1790
1790 20 IF $DFA0 = 177 THEN GOTO 1800
1800 20 IF $DFA0 = 178 THEN GOTO 1810
1810 20 IF $DFA0 = 179 THEN GOTO 1820
1820 20 IF $DFA0 = 180 THEN GOTO 1830
1830 20 IF $DFA0 = 181 THEN GOTO 1840
1840 20 IF $DFA0 = 182 THEN GOTO 1850
1850 20 IF $DFA0 = 183 THEN GOTO 1860
1860 20 IF $DFA0 = 184 THEN GOTO 1870
1870 20 IF $DFA0 = 185 THEN GOTO 1880
1880 20 IF $DFA0 = 186 THEN GOTO 1890
1890 20 IF $DFA0 = 187 THEN GOTO 1900
1900 20 IF $DFA0 = 188 THEN GOTO 1910
1910 20 IF $DFA0 = 189 THEN GOTO 1920
1920 20 IF $DFA0 = 190 THEN GOTO 1930
1930 20 IF $DFA0 = 191 THEN GOTO 1940
1940 20 IF $DFA0 = 192 THEN GOTO 1950
1950 20 IF $DFA0 = 193 THEN GOTO 1960
1960 20 IF $DFA0 = 194 THEN GOTO 1970
1970 20 IF $DFA0 = 195 THEN GOTO 1980
1980 20 IF $DFA0 = 196 THEN GOTO 1990
1990 20 IF $DFA0 = 197 THEN GOTO 2000
2000 20 IF $DFA0 = 198 THEN GOTO 2010
2010 20 IF $DFA0 = 199 THEN GOTO 2020
2020 20 IF $DFA0 = 200 THEN GOTO 2030
2030 20 IF $DFA0 = 201 THEN GOTO 2040
2040 20 IF $DFA0 = 202 THEN GOTO 2050
2050 20 IF $DFA0 = 203 THEN GOTO 2060
2060 20 IF $DFA0 = 204 THEN GOTO 2070
2070 20 IF $DFA0 = 205 THEN GOTO 2080
2080 20 IF $DFA0 = 206 THEN GOTO 2090
2090 20 IF $DFA0 = 207 THEN GOTO 2100
2100 20 IF $DFA0 = 208 THEN GOTO 2110
2110 20 IF $DFA0 = 209 THEN GOTO 2120
2120 20 IF $DFA0 = 210 THEN GOTO 2130
2130 20 IF $DFA0 = 211 THEN GOTO 2140
2140 20 IF $DFA0 = 212 THEN GOTO 2150
2150 20 IF $DFA0 = 213 THEN GOTO 2160
2160 20 IF $DFA0 = 214 THEN GOTO 2170
2170 20 IF $DFA0 = 215 THEN GOTO 2180
2180 20 IF $DFA0 = 216 THEN GOTO 2190
2190 20 IF $DFA0 = 217 THEN GOTO 2200
2200 20 IF $DFA0 = 218 THEN GOTO 2210
2210 20 IF $DFA0 = 219 THEN GOTO 2220
2220 20 IF $DFA0 = 220 THEN GOTO 2230
2230 20 IF $DFA0 = 221 THEN GOTO 2240
2240 20 IF $DFA0 = 222 THEN GOTO 2250
2250 20 IF $DFA0 = 223 THEN GOTO 2260
2260 20 IF $DFA0 = 224 THEN GOTO 2270
2270 20 IF $DFA0 = 225 THEN GOTO 2280
2280 20 IF $DFA0 = 226 THEN GOTO 2290
2290 20 IF $DFA0 = 227 THEN GOTO 2300
2300 20 IF $DFA0 = 228 THEN GOTO 2310
2310 20 IF $DFA0 = 229 THEN GOTO 2320
2320 20 IF $DFA0 = 230 THEN GOTO 2330
2330 20 IF $DFA0 = 231 THEN GOTO 2340
2340 20 IF $DFA0 = 232 THEN GOTO 2350
2350 20 IF $DFA0 = 233 THEN GOTO 2360
2360 20 IF $DFA0 = 234 THEN GOTO 2370
2370 20 IF $DFA0 = 235 THEN GOTO 2380
2380 20 IF $DFA0 = 236 THEN GOTO 2390
2390 20 IF $DFA0 = 237 THEN GOTO 2400
2400 20 IF $DFA0 = 238 THEN GOTO 2410
2410 20 IF $DFA0 = 239 THEN GOTO 2420
2420 20 IF $DFA0 = 240 THEN GOTO 2430
2430 20 IF $DFA0 = 241 THEN GOTO 2440
2440 20 IF $DFA0 = 242 THEN GOTO 2450
2450 20 IF $DFA0 = 243 THEN GOTO 2460
2460 20 IF $DFA0 = 244 THEN GOTO 2470
2470 20 IF $DFA0 = 245 THEN GOTO 2480
2480 20 IF $DFA0 = 246 THEN GOTO 2490
2490 20 IF $DFA0 = 247 THEN GOTO 2500
2500 20 IF $DFA0 = 248 THEN GOTO 2510
2510 20 IF $DFA0 = 249 THEN GOTO 2520
2520 20 IF $DFA0 = 250 THEN GOTO 2530
2530 20 IF $DFA0 = 251 THEN GOTO 2540
2540 20 IF $DFA0 = 252 THEN GOTO 2550
2550 20 IF $DFA0 = 253 THEN GOTO 2560
2560 20 IF $DFA0 = 254 THEN GOTO 2570
2570 20 IF $DFA0 = 255 THEN GOTO 2580
2580 20 IF $DFA0 = 256 THEN GOTO 2590
2590 20 IF $DFA0 = 257 THEN GOTO 2600
2600 20 IF $DFA0 = 258 THEN GOTO 2610
2610 20 IF $DFA0 = 259 THEN GOTO 2620
2620 20 IF $DFA0 = 260 THEN GOTO 2630
2630 20 IF $DFA0 = 261 THEN GOTO 2640
2640 20 IF $DFA0 = 262 THEN GOTO 2650
2650 20 IF $DFA0 = 263 THEN GOTO 2660
2660 20 IF $DFA0 = 264 THEN GOTO 2670
2670 20 IF $DFA0 = 265 THEN GOTO 2680
2680 20 IF $DFA0 = 266 THEN GOTO 2690
2690 20 IF $DFA0 = 267 THEN GOTO 2700
2700 20 IF $DFA0 = 268 THEN GOTO 2710
2710 20 IF $DFA0 = 269 THEN GOTO 2720
2720 20 IF $DFA0 = 270 THEN GOTO 2730
2730 20 IF $DFA0 = 271 THEN GOTO 2740
2740 20 IF $DFA0 = 272 THEN GOTO 2750
2750 20 IF $DFA0 = 273 THEN GOTO 2760
2760 20 IF $DFA0 = 274 THEN GOTO 2770
2770 20 IF $DFA0 = 275 THEN GOTO 2780
2780 20 IF $DFA0 = 276 THEN GOTO 2790
2790 20 IF $DFA0 = 277 THEN GOTO 2800
2800 20 IF $DFA0 = 278 THEN GOTO 2810
2810 20 IF $DFA0 = 279 THEN GOTO 2820
2820 20 IF $DFA0 = 280 THEN GOTO 2830
2830 20 IF $DFA0 = 281 THEN GOTO 2840
2840 20 IF $DFA0 = 282 THEN GOTO 2850
2850 20 IF $DFA0 = 283 THEN GOTO 2860
2860 20 IF $DFA0 = 284 THEN GOTO 2870
2870 20 IF $DFA0 = 285 THEN GOTO 2880
2880 20 IF $DFA0 = 286 THEN GOTO 2890
2890 20 IF $DFA0 = 287 THEN GOTO 2900
2900 20 IF $DFA0 = 288 THEN GOTO 2910
2910 20 IF $DFA0 = 289 THEN GOTO 2920
2920 20 IF $DFA0 = 290 THEN GOTO 2930
2930 20 IF $DFA0 = 291 THEN GOTO 2940
2940 20 IF $DFA0 = 292 THEN GOTO 2950
2950 20 IF $DFA0 = 293 THEN GOTO 2960
2960 20 IF $DFA0 = 294 THEN GOTO 2970
2970 20 IF $DFA0 = 295 THEN GOTO 2980
2980 20 IF $DFA0 = 296 THEN GOTO 2990
2990 20 IF $DFA0 = 297 THEN GOTO 3000
3000 20 IF $DFA0 = 298 THEN GOTO 3010
3010 20 IF $DFA0 = 299 THEN GOTO 3020
3020 20 IF $DFA0 = 300 THEN GOTO 3030
3030 20 IF $DFA0 = 301 THEN GOTO 3040
3040 20 IF $DFA0 = 302 THEN GOTO 3050
3050 20 IF $DFA0 = 303 THEN GOTO 3060
3060 20 IF $DFA0 = 304 THEN GOTO 3070
3070 20 IF $DFA0 = 305 THEN GOTO 3080
3080 20 IF $DFA0 = 306 THEN GOTO 3090
3090 20 IF $DFA0 = 307 THEN GOTO 3100
3100 20 IF $DFA0 = 308 THEN GOTO 3110
3110 20 IF $DFA0 = 309 THEN GOTO 3120
3120 20 IF $DFA0 = 310 THEN GOTO 3130
3130 20 IF $DFA0 = 311 THEN GOTO 3140
3140 20 IF $DFA0 = 312 THEN GOTO 3150
3150 20 IF $DFA0 = 313 THEN GOTO 3160
3160 20 IF $DFA0 = 314 THEN GOTO 3170
3170 20 IF $DFA0 = 315 THEN GOTO 3180
3180 20 IF $DFA0 = 316 THEN GOTO 3190
3190 20 IF $DFA0 = 317 THEN GOTO 3200
3200 20 IF $DFA0 = 318 THEN GOTO 3210
3210 20 IF $DFA0 = 319 THEN GOTO 3220
3220 20 IF $DFA0 = 320 THEN GOTO 3230
3230 20 IF $DFA0 = 321 THEN GOTO 3240
3240 20 IF $DFA0 = 322 THEN GOTO 3250
3250 20 IF $DFA0 = 323 THEN GOTO 3260
3260 20 IF $DFA0 = 324 THEN GOTO 3270
3270 20 IF $DFA0 = 325 THEN GOTO 3280
3280 20 IF $DFA0 = 326 THEN GOTO 3290
3290 20 IF $DFA0 = 327 THEN GOTO 3300
3300 20 IF $DFA0 = 328 THEN GOTO 3310
3310 20 IF $DFA0 = 329 THEN GOTO 3320
3320 20 IF $DFA0 = 330 THEN GOTO 3330
3330 20 IF $DFA0 = 331 THEN GOTO 3340
3340 20 IF $DFA0 = 332 THEN GOTO 3350
3350 20 IF $DFA0 = 333 THEN GOTO 3360
3360 20 IF $DFA0 = 334 THEN GOTO 3370
3370 20 IF $DFA0 = 335 THEN GOTO 3380
3380 20 IF $DFA0 = 336 THEN GOTO 3390
3390 20 IF $DFA0 = 337 THEN GOTO 3400
3400 20 IF $DFA0 = 338 THEN GOTO 3410
3410 20 IF $DFA0 = 339 THEN GOTO 3420
3420 20 IF $DFA0 = 340 THEN GOTO 3430
3430 20 IF $DFA0 = 341 THEN GOTO 3440
3440 20 IF $DFA0 = 342 THEN GOTO 3450
3450 20 IF $DFA0 = 343 THEN GOTO 3460
3460 20 IF $DFA0 = 344 THEN GOTO 3470
3470 20 IF $DFA0 = 345 THEN GOTO 3480
3480 20 IF $DFA0 = 346 THEN GOTO 3490
3490 20 IF $DFA0 = 347 THEN GOTO 3500
3500 20 IF $DFA0 = 348 THEN GOTO 3510
3510 20 IF $DFA0 = 349 THEN GOTO 3520
3520 20 IF $DFA0 = 350 THEN GOTO 3530
3530 20 IF $DFA0 = 351 THEN GOTO 3540
3540 20 IF $DFA0 = 352 THEN GOTO 3550
3550 20 IF $DFA0 = 353 THEN GOTO 3560
3560 20 IF $DFA0 = 354 THEN GOTO 3570
3570 20 IF $DFA0 = 355 THEN GOTO 3580
3580 20 IF $DFA0 = 356 THEN GOTO 3590
3590 20 IF $DFA0 = 357 THEN GOTO 3600
3600 20 IF $DFA0 = 358 THEN GOTO 3610
3610 20 IF $DFA0 = 359 THEN GOTO 3620
3620 20 IF $DFA0 = 360 THEN GOTO 3630
3630 20 IF $DFA0 = 361 THEN GOTO 3640
3640 20 IF $DFA0 = 362 THEN GOTO 3650
3650 20 IF $DFA0 = 363 THEN GOTO 3660
3660 20 IF $DFA0 = 364 THEN GOTO 3670
3670 20 IF $DFA0 = 365 THEN GOTO 3680
3680 20 IF $DFA0 = 366 THEN GOTO 3690
3690 20 IF $DFA0 = 367 THEN GOTO 3700
3700 20 IF $DFA0 = 368 THEN GOTO 3710
3710 20 IF $DFA0 = 369 THEN GOTO 3720
3720 20 IF $DFA0 = 370 THEN GOTO 3730
3730 20 IF $DFA0 = 371 THEN GOTO 3740
3740 20 IF $DFA0 = 372 THEN GOTO 3750
3750 20 IF $DFA0 = 373 THEN GOTO 3760
3760 20 IF $DFA0 = 374 THEN GOTO 3770
3770 20 IF $DFA0 = 375 THEN GOTO 3780
3780 20 IF $DFA0 = 376 THEN GOTO 3790
3790 20 IF $DFA0 = 377 THEN GOTO 3800
3800 20 IF $DFA0 = 378 THEN GOTO 3810
3810 20 IF $DFA0 = 379 THEN GOTO 3820
3820 20 IF $DFA0 = 380 THEN GOTO 3830
3830 20 IF $DFA0 = 381 THEN GOTO 3840
3840 20 IF $DFA0 = 382 THEN GOTO 3850
3850 20 IF $DFA0 = 383 THEN GOTO 3860
3860 20 IF $DFA0 = 384 THEN GOTO 3870
3870 20 IF $DFA0 = 385 THEN GOTO 3880
3880 20 IF $DFA0 = 386 THEN GOTO 3890
3890 20 IF $DFA0 = 387 THEN GOTO 3900
3900 20 IF $DFA0 = 388 THEN GOTO 3910
3910 20 IF $DFA0 = 389 THEN GOTO 3920
3920 20 IF $DFA0 = 390 THEN GOTO 3930
3930 20 IF $DFA0 = 391 THEN GOTO 3940
3940 20 IF $DFA0 = 392 THEN GOTO 3950
3950 20 IF $DFA0 = 393 THEN GOTO 3960
3960 20 IF $DFA0 = 394 THEN GOTO 3970
3970 20 IF $DFA0 = 395 THEN GOTO 3980
3980 20 IF $DFA0 = 396 THEN GOTO 3990
3990 20 IF $DFA0 = 397 THEN GOTO 4000
4000 20 IF $DFA0 = 398 THEN GOTO 4010
4010 20 IF $DFA0 = 399 THEN GOTO 4020
4020 20 IF $DFA0 = 400 THEN GOTO 4030
4030 20 IF $DFA0 = 401 THEN GOTO 4040
4040 20 IF $DFA0 = 402 THEN GOTO 4050
4050 20 IF $DFA0 = 403 THEN GOTO 4060
4060 20 IF $DFA0 = 404 THEN GOTO 4070
4070 20 IF $DFA0 = 405 THEN GOTO 4080
4080 20 IF $DFA0 = 406 THEN GOTO 4090
4090 20 IF $DFA0 = 407 THEN GOTO 4100
4100 20 IF $DFA0 = 408 THEN GOTO 4110
4110 20 IF $DFA0 = 409 THEN GOTO 4120
4120 20 IF $DFA0 = 410 THEN GOTO 4130
4130 20 IF $DFA0 = 411 THEN GOTO 4140
4140 20 IF $DFA0 = 412 THEN GOTO 4150
4150 20 IF $DFA0 = 413 THEN GOTO 4
```

YOUR COMMISSION July 1999

27	1690 DATA 1 88 82 138 88 82	44	2190 DATA 58 44 72 48 88 32	67	3690 DATA 88 188 78 78 87 78
28	1700 DATA 88 88 88 88 88 88	45	2200 DATA 88 88 88 88 88 88	68	3700 DATA 72 88 88 88 88 88
29	1710 DATA 88 88 88 88 88 88	46	2210 DATA 88 88 88 88 88 88	69	3710 DATA 88 88 88 88 88 88
30	1720 DATA 88 88 88 88 88 88	47	2220 DATA 88 88 88 88 88 88	70	3720 DATA 88 88 88 88 88 88
31	1730 DATA 88 88 88 88 88 88	48	2230 DATA 88 88 88 88 88 88	71	3730 DATA 88 88 88 88 88 88
32	1740 DATA 88 88 88 88 88 88	49	2240 DATA 88 88 88 88 88 88	72	3740 DATA 88 88 88 88 88 88
33	1750 DATA 88 88 88 88 88 88	50	2250 DATA 88 88 88 88 88 88	73	3750 DATA 88 88 88 88 88 88
34	1760 DATA 88 88 88 88 88 88	51	2260 DATA 88 88 88 88 88 88	74	3760 DATA 88 88 88 88 88 88
35	1770 DATA 88 88 88 88 88 88	52	2270 DATA 88 88 88 88 88 88	75	3770 DATA 88 88 88 88 88 88
36	1780 DATA 88 88 88 88 88 88	53	2280 DATA 88 88 88 88 88 88	76	3780 DATA 88 88 88 88 88 88
37	1790 DATA 88 88 88 88 88 88	54	2290 DATA 88 88 88 88 88 88	77	3790 DATA 88 88 88 88 88 88
38	1800 DATA 88 88 88 88 88 88	55	2300 DATA 88 88 88 88 88 88	78	3800 DATA 88 88 88 88 88 88
39	1810 DATA 88 88 88 88 88 88	56	2310 DATA 88 88 88 88 88 88	79	3810 DATA 88 88 88 88 88 88
40	1820 DATA 88 88 88 88 88 88	57	2320 DATA 88 88 88 88 88 88	80	3820 DATA 88 88 88 88 88 88
41	1830 DATA 88 88 88 88 88 88	58	2330 DATA 88 88 88 88 88 88	81	3830 DATA 88 88 88 88 88 88
42	1840 DATA 88 88 88 88 88 88	59	2340 DATA 88 88 88 88 88 88	82	3840 DATA 88 88 88 88 88 88
43	1850 DATA 88 88 88 88 88 88	60	2350 DATA 88 88 88 88 88 88	83	3850 DATA 88 88 88 88 88 88
44	1860 DATA 88 88 88 88 88 88	61	2360 DATA 88 88 88 88 88 88	84	3860 DATA 88 88 88 88 88 88
45	1870 DATA 88 88 88 88 88 88	62	2370 DATA 88 88 88 88 88 88	85	3870 DATA 88 88 88 88 88 88
46	1880 DATA 88 88 88 88 88 88	63	2380 DATA 88 88 88 88 88 88	86	3880 DATA 88 88 88 88 88 88
47	1890 DATA 88 88 88 88 88 88	64	2390 DATA 88 88 88 88 88 88	87	3890 DATA 88 88 88 88 88 88
48	1900 DATA 88 88 88 88 88 88	65	2400 DATA 88 88 88 88 88 88	88	3900 DATA 88 88 88 88 88 88
49	1910 DATA 88 88 88 88 88 88	66	2410 DATA 88 88 88 88 88 88	89	3910 DATA 88 88 88 88 88 88
50	1920 DATA 88 88 88 88 88 88	67	2420 DATA 88 88 88 88 88 88	90	3920 DATA 88 88 88 88 88 88
51	1930 DATA 88 88 88 88 88 88	68	2430 DATA 88 88 88 88 88 88	91	3930 DATA 88 88 88 88 88 88
52	1940 DATA 88 88 88 88 88 88	69	2440 DATA 88 88 88 88 88 88	92	3940 DATA 88 88 88 88 88 88
53	1950 DATA 88 88 88 88 88 88	70	2450 DATA 88 88 88 88 88 88	93	3950 DATA 88 88 88 88 88 88
54	1960 DATA 88 88 88 88 88 88	71	2460 DATA 88 88 88 88 88 88	94	3960 DATA 88 88 88 88 88 88
55	1970 DATA 88 88 88 88 88 88	72	2470 DATA 88 88 88 88 88 88	95	3970 DATA 88 88 88 88 88 88
56	1980 DATA 88 88 88 88 88 88	73	2480 DATA 88 88 88 88 88 88	96	3980 DATA 88 88 88 88 88 88
57	1990 DATA 88 88 88 88 88 88	74	2490 DATA 88 88 88 88 88 88	97	3990 DATA 88 88 88 88 88 88
58	2000 DATA 88 88 88 88 88 88	75	2500 DATA 88 88 88 88 88 88	98	4000 DATA 88 88 88 88 88 88
59	2010 DATA 88 88 88 88 88 88	76	2510 DATA 88 88 88 88 88 88	99	4010 DATA 88 88 88 88 88 88
60	2020 DATA 88 88 88 88 88 88	77	2520 DATA 88 88 88 88 88 88	100	4020 DATA 88 88 88 88 88 88
61	2030 DATA 88 88 88 88 88 88	78	2530 DATA 88 88 88 88 88 88		
62	2040 DATA 88 88 88 88 88 88	79	2540 DATA 88 88 88 88 88 88		
63	2050 DATA 88 88 88 88 88 88	80	2550 DATA 88 88 88 88 88 88		
64	2060 DATA 88 88 88 88 88 88	81	2560 DATA 88 88 88 88 88 88		
65	2070 DATA 88 88 88 88 88 88	82	2570 DATA 88 88 88 88 88 88		
66	2080 DATA 88 88 88 88 88 88	83	2580 DATA 88 88 88 88 88 88		
67	2090 DATA 88 88 88 88 88 88	84	2590 DATA 88 88 88 88 88 88		
68	2100 DATA 88 88 88 88 88 88	85	2600 DATA 88 88 88 88 88 88		
69	2110 DATA 88 88 88 88 88 88	86	2610 DATA 88 88 88 88 88 88		
70	2120 DATA 88 88 88 88 88 88	87	2620 DATA 88 88 88 88 88 88		
71	2130 DATA 88 88 88 88 88 88	88	2630 DATA 88 88 88 88 88 88		
72	2140 DATA 88 88 88 88 88 88	89	2640 DATA 88 88 88 88 88 88		
73	2150 DATA 88 88 88 88 88 88	90	2650 DATA 88 88 88 88 88 88		
74	2160 DATA 88 88 88 88 88 88	91	2660 DATA 88 88 88 88 88 88		
75	2170 DATA 88 88 88 88 88 88	92	2670 DATA 88 88 88 88 88 88		
76	2180 DATA 88 88 88 88 88 88	93	2680 DATA 88 88 88 88 88 88		
77	2190 DATA 88 88 88 88 88 88	94	2690 DATA 88 88 88 88 88 88		
78	2200 DATA 88 88 88 88 88 88	95	2700 DATA 88 88 88 88 88 88		
79	2210 DATA 88 88 88 88 88 88	96	2710 DATA 88 88 88 88 88 88		
80	2220 DATA 88 88 88 88 88 88	97	2720 DATA 88 88 88 88 88 88		
81	2230 DATA 88 88 88 88 88 88	98	2730 DATA 88 88 88 88 88 88		
82	2240 DATA 88 88 88 88 88 88	99	2740 DATA 88 88 88 88 88 88		
83	2250 DATA 88 88 88 88 88 88	100	2750 DATA 88 88 88 88 88 88		
84	2260 DATA 88 88 88 88 88 88				
85	2270 DATA 88 88 88 88 88 88				
86	2280 DATA 88 88 88 88 88 88				
87	2290 DATA 88 88 88 88 88 88				
88	2300 DATA 88 88 88 88 88 88				
89	2310 DATA 88 88 88 88 88 88				
90	2320 DATA 88 88 88 88 88 88				
91	2330 DATA 88 88 88 88 88 88				
92	2340 DATA 88 88 88 88 88 88				
93	2350 DATA 88 88 88 88 88 88				
94	2360 DATA 88 88 88 88 88 88				
95	2370 DATA 88 88 88 88 88 88				
96	2380 DATA 88 88 88 88 88 88				
97	2390 DATA 88 88 88 88 88 88				
98	2400 DATA 88 88 88 88 88 88				
99	2410 DATA 88 88 88 88 88 88				
100	2420 DATA 88 88 88 88 88 88				

60	0000 DATA 11 34 184 34 33 13	13	0000 DATA 58 59 172 87 79 78	70	0000 DATA 69 87 68 177 65 17
70	0000 DATA 30 334 1 334 44 54	80	0000 DATA 88 87 87 88 44 88	80	0000 DATA 87 87 88 179 48 84
80	0000 DATA 58 58 183 183 34 38	90	0000 DATA 88 88 88 33 83 83	90	0000 DATA 88 88 88 178 48 83
90	0000 DATA 88 134 78 134 88 8	00	0000 DATA 30 33 33 33 33 33	00	0000 DATA 88 88 714 98 88 88
00	0000 DATA 58 133 84 133 84 8	10	0000 DATA 74 74 74 88 8 74 74	10	0000 DATA 10 10 10 10 10 10
10	0000 DATA 36 364 1 364 44 34	20	0000 DATA 8 88 8 88 44 88 8	20	0000 DATA 118 88 47 178 33 1
20	0000 DATA 48 54 58 183 183 4	30	0000 DATA 88 88 44 34 88 88	30	0000 DATA 87 177 33 88 137 3
30	0000 DATA 43 34 38 33 88 88	40	0000 DATA 34 38 334 44 34 37	40	0000 DATA 8 387 38 188 1 188
40	0000 DATA 58 58 32 32 48 38	50	0000 DATA 58 58 133 88 87 8	50	0000 DATA 87 88 88 178 48 83
50	0000 DATA 37 384 1 384 44 34	60	0000 DATA 118 8 138 44 33 33	60	0000 DATA 88 88 88 88 88 88
60	0000 DATA 58 58 38 183 183 38	70	0000 DATA 88 44 34 33 33 33	70	0000 DATA 187 83 88 87 88 88
70	0000 DATA 54 54 48 27 83 1 338	80	0000 DATA 133 87 87 8 143 38	80	0000 DATA 88 88 88 88 88 88
80	0000 DATA 54 44 84 84 84 84	90	0000 DATA 188 87 88 87 88 88	90	0000 DATA 83 84 87 83 8 138
90	0000 DATA 214 44 81 38 33 4	00	0000 DATA 8 88 88 88 88 88	00	0000 DATA 1 138 88 88 88 88
00	0000 DATA 88 44 33 33 88	10	0000 DATA 58 58 178 88 17	10	0000 DATA 48 54 147 88 88 88
10	0000 DATA 113 37 38 3 338 44	20	0000 DATA 138 34 138 3 134 4	20	0000 DATA 138 48 8 48 378
20	0000 DATA 44 84 88 88 178 84	30	0000 DATA 34 38 34 144 34 33	30	0000 DATA 48 54 58 48 88 88
30	0000 DATA 314 38 88 47 88 88	40	0000 DATA 58 58 138 44 84 84	40	0000 DATA 88 88 88 88 88 88
40	0000 DATA 44 33 38 48 34 88	50	0000 DATA 51 58 138 88 88 88	50	0000 DATA 48 57 137 34 48 88
50	0000 DATA 58 58 33 33 88 88	60	0000 DATA 33 33 33 33 33 33	60	0000 DATA 138 88 87 177 48 8
60	0000 DATA 143 88 88 48 48	70	0000 DATA 33 33 33 143 33 33	70	0000 DATA 77 48 8 83 48 338
70	0000 DATA 43 48 8 138 37 88	80	0000 DATA 77 78 84 33 84 78	80	0000 DATA 88 87 178 48 84 13
80	0000 DATA 348 57 84 3 47 7	90	0000 DATA 13 38 148 3 134 44	90	0000 DATA 88 58 113 188 88 8
90	0000 DATA 178 48 38 88 88 88	00	0000 DATA 44 48 31 44 34 38	00	0000 DATA 87 137 33 87 48 8
00	0000 DATA 38 84 48 178 78 88	10	0000 DATA 34 38 334 44 33 33	10	0000 DATA 348 3 88 88 78 177
10	0000 DATA 58 134 3 44 78 33	20	0000 DATA 178 38 38 88 88 88	20	0000 DATA 87 88 88 88 88 88
20	0000 DATA 33 33 38 47 33 73	30	0000 DATA 88 48 8 88 34 184	30	0000 DATA 33 33 33 33 33 33
30	0000 DATA 33 33 83 33 43 83	40	0000 DATA 88 47 178 48 54 13	40	0000 DATA 33 37 33 33 33 33
40	0000 DATA 78 73 84 8 113 37	50	0000 DATA 48 54 113 138 88 8	50	0000 DATA 33 33 33 33 33 33
50	0000 DATA 84 8 133 37 88 8	60	0000 DATA 54 137 88 33 48 8	60	0000 DATA 147 88 88 88 88 88
60	0000 DATA 48 48 33 88 88	70	0000 DATA 148 8 138 88 47 17	70	0000 DATA 88 88 88 88 88 88
70	0000 DATA 33 87 88 3 143 8	80	0000 DATA 133 34 48 48 88 81	80	0000 DATA 88 88 88 88 88 88
80	0000 DATA 88 133 34 138 84	90	0000 DATA 87 177 88 88 137 8	90	0000 DATA 88 88 88 88 88 88
90	0000 DATA 138 88 88 88 88 88	00	0000 DATA 8 138 34 178 3 138	00	0000 DATA 88 88 88 88 88 88

YOUR CONSIDERATION ONLY

1000

FIRST micro

AMIGA
HARDWARE & SOFTWARE
SPECIALISTS

8 FORD CLOSE, BRILL, MK5 8ET
VERMOREL, DORSET



INC VAT

0202 813176



FREE DELIVERY

AMIGA A500

+ FREE £220 SOFTWARE

INCLUDING - ● PURPLE SATURN ● HOSTAGES
● INTERNATIONAL SOCCER ● WINTER OLYMPIAD
● SPLITTING IMAGE ● STARGOOSE ● BACKLASH
● QUADRANGLE ● ELIMINATOR
● FANTASTICK F3 JOYSTICK
● TV MODULATOR

£389 inc

AMIGA A500/ PHILIPS 8833

£599 inc

AMIGA B2000/ PHILIPS 8833

INC ACCESSORIES

£1199

AMIGA A500

INCLUDING -
MOUSE WORKBENCH
UTILITIES MANUALS
BASIC TUTORIAL
TV MODULATOR

£305 + VAT

AMIGA B2000/ PHILIPS 8833

INCLUDING -
MOUSE WORKBENCH BASIC UTILITIES
MANUALS ● XT BRIDGE BOARD
20Mb HARD DISC

£1599 inc

AMIGA B2000

INCLUDING - MOUSE WORKBENCH
BASIC UTILITIES MANUALS

£999

A300 HARDWARE

A300 + TV CARD £299
A300 + CPU BOARD £299
A300 + MOUSE BOARD £299
A300 + 20Mb DISC £299
A300 + 40Mb DISC £299

AMIGA EXTERNAL DRIVES

Commodore Disk II 128 £59
Disk II 256 £59
Commodore Disk II 500 £59
Disk II 500 + 20Mb DISC £59

AMIGA ACCESSORIES

Amiga A500 £299
TV Modulator £299
Mouse Board £299
Amiga Disk II 500 £299
Amiga Disk II 500 + 20Mb DISC £299

COMMODORE C64

C64 £299
C64 + TV CARD £299
C64 + CPU BOARD £299
C64 + 20Mb DISC £299
C64 + 40Mb DISC £299

A2000 HARDWARE

A2000 + TV CARD £299
A2000 + CPU BOARD £299
A2000 + 20Mb DISC £299
A2000 + 40Mb DISC £299
A2000 + 80Mb DISC £299

AMIGA B2000 MONITORS

Amiga B2000 128 £59
Amiga B2000 256 £59
Amiga B2000 500 £59
Amiga B2000 500 + 20Mb DISC £59

PRINTERS

Amiga A500 £299
Amiga A500 + CPU BOARD £299
Amiga A500 + 20Mb DISC £299
Amiga A500 + 40Mb DISC £299
Amiga A500 + 80Mb DISC £299

COLOR PRINTERS

Amiga A500 £299
Amiga A500 + CPU BOARD £299
Amiga A500 + 20Mb DISC £299
Amiga A500 + 40Mb DISC £299
Amiga A500 + 80Mb DISC £299

LASERS

Amiga A500 £299
Amiga A500 + CPU BOARD £299
Amiga A500 + 20Mb DISC £299
Amiga A500 + 40Mb DISC £299
Amiga A500 + 80Mb DISC £299

BUSINESS & MORE

Amiga A500 £299
Amiga A500 + CPU BOARD £299
Amiga A500 + 20Mb DISC £299
Amiga A500 + 40Mb DISC £299
Amiga A500 + 80Mb DISC £299

Amiga A500 £299
Amiga A500 + CPU BOARD £299
Amiga A500 + 20Mb DISC £299
Amiga A500 + 40Mb DISC £299
Amiga A500 + 80Mb DISC £299

UK TOP 10 GAMES

1. Fantasy £19.99
2. Wizard of Witches £19.99
3. Dragon Master £19.99
4. Castle Master £19.99
5. The Wizard of Witches £19.99
6. Dragon Master £19.99
7. Fantasy £19.99
8. The Wizard of Witches £19.99
9. Dragon Master £19.99
10. Castle Master £19.99
11. The Wizard of Witches £19.99
12. Dragon Master £19.99
13. Fantasy £19.99
14. The Wizard of Witches £19.99
15. Dragon Master £19.99
16. Castle Master £19.99
17. The Wizard of Witches £19.99
18. Dragon Master £19.99
19. Fantasy £19.99
20. The Wizard of Witches £19.99

ALL PRICES INCLUDE VAT

CREDIT CARD MAIL ORDER
AND EXPORT HOTLINE

0202 813176

OPEN MON-SAT 9am-5.30pm

To: First Micro, 8 Ford Close, Brill, MK5 8ET, Vermorel, Dorset

My computer is
Amiga A500/A500+ for £
or Amiga B2000 for £

Signature

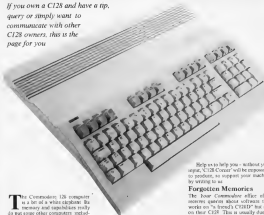
Address

Postcode

For info

C128 Corner

If you own a C128 and have a tip, query or simply want to communicate with other C128 owners, this is the page for you



The Commodore 128 computer is a bit of a white elephant. Its memory and capabilities really do put some other computers, including the C64 and C16, to shame. Not only does it offer total compatibility with the extremely popular C64, but it also boasts an impressive lineup, both 40 and 80 column screen formats and a massive 128K memory (not as much as an Amiga, but double that of a C64).

Not only do the facilities of the machine itself far surpass those of the C64, but its disk drives are superior to the lumbering 1541's. Both the 1570 and 1571 are far prettier than the 1541 drive, but the 1571 is also double-sided, giving double the amount of storage per disk—without having to resort to the flippage technique of turning the disk over.

Unfortunately though, the similarities between the C128 and the white elephant do not end with the size and power of the computer. Software for the C128 is extremely scarce, and the C128 computer itself seems to be about

as common as a white elephant.

It's a great shame that the C128 was given the acclaim it deserved by the software buyers, or even Commodore itself. Your Commodore has been one of the few magazines that has continuously supported the C128 range over the years, and this page is simply an extension of that support.

C128 Corner is the page for all Commodore 128 users. It's designed to be a forum for all queries and tips relating to your computer. But, and it's a big but, "C128 Corner" cannot succeed without your help. If you have a query about your 128 or software for the machine, then write us this page. If you have a tip you'd like to share with other C128 users, write to us. If you learn about anything happening in the C128 world that hasn't been covered in the magazine, write to us.

Help us to help you—without your input, "C128 Corner" will be impossible to produce, so support your machine by writing to us.

Forgotten Memories

The four Commodore editors often receive queries about software that works on "a friend's C128D" but not on their C128. This is usually due to the fact that the C128 and C128D computers are not totally identical: the C128D computer has more video memory than most (but not all) standard C128 computers. The difference only becomes apparent if you're running a program that uses the extra 64K of video RAM that is present on some 128s.

The problem is not insurmountable. All that's required is a memory upgrade for your C128 that gives your computer access to 64K extra video RAM. Functional Systems Software is the only company we're aware of that sells such a memory upgrade. It's a small circuit board that's plugged into the board inside your C128.

As long as you can use a screwdriver and are capable of delicate work, you should have no problems doing the board yourself. Don't forget though, that if you do open up your

Quick Search

*If you need to change
anything in your Basic text,
find and list it
with this handy routine
By Neil Higgins*



How many times have you been programming in Basic, and suddenly found out that you need to change a variable to another value, or re-write a string of text? Put it this way - if you'd a pound for every time you've had to re-write a Basic program, you'd be richer than the ruler of Your Commodore!

The main problem encountered is that you always need to search the whole program to make sure all the correct changes have been made, especially if it's a variable that needs changing.

This all adds up to lots of typing of the command LIST, which in a long program can be a bit of a pain. Well that's no more, the routine given here will do all the searching and listing for you, and all you need supply is the text to be searched for. To do this, the syntax is

SYNOPSIS, Text

For example, if you wanted to list all the lines in your program where the variable A\$ appears, you would simply enter:-

SYNOPSIS, A\$

Keywords can also be searched for. If you wanted to list all REM statements, you'd enter:-

SYNOPSIS, REM

As with all keywords, to save a lot of typing you can enter the abbreviated form. For example, to search for all occurrences of the PRINT statement you may use:-

SYNOPSIS, P

If you wanted to search for a full string, that's how you would enter it. For example, to find a string defined as "My word", you'd enter:-

SYNOPSIS, "My word"

Note that the same string would also be used if you entered:-

SYNOPSIS, My word.

But so would any other line containing the words, My word, if you get my point?

If you experiment with a large Basic program, you'll find out the correct way to use the routine. Sometimes, if the character appears in a lot of lines, they will disappear

off the top of the screen like a normal listing. To prevent this, you can use the space bar to pause the listing in which case pressing another key will continue the search. The Run/stop key will break out. Please note: this routine can only be called in direct mode, not from within a Basic program.

For the technically minded, the routine resides in the 4K block of Ram from \$C000, and stores the text to be searched for in a table directly after itself. Machine code will find the source code useful, particularly if they want to conclude the routine as part of an extended Basic type program. The routine was written using the Four Commodore Speedy Assembler, but should be transplantable to most assemblers.

Getting It All In

I've supplied two methods of typing in the program. Method one provides the source listing if you have an assembler. This listing is well documented, so you can see how the routine works. Secondly, I have provided a Basic loader. Remember to save your efforts before testing it - accidents will happen.

PROGRAM LISTING SEARCH		SYNOPSIS, A\$		SYNOPSIS, REM	
61	100 REM *****	11	100 REM *****	21	100 REM *****
62	110 REM *****	12	110 REM *****	22	110 REM *****
63	120 REM *****	13	120 REM *****	23	120 REM *****
64	130 REM *****	14	130 REM *****	24	130 REM *****
65	140 REM *****	15	140 REM *****	25	140 REM *****
66	150 REM *****	16	150 REM *****	26	150 REM *****
67	160 REM *****	17	160 REM *****	27	160 REM *****
68	170 REM *****	18	170 REM *****	28	170 REM *****
69	180 REM *****	19	180 REM *****	29	180 REM *****
70	190 REM *****	20	190 REM *****	30	190 REM *****
71	200 REM *****	21	200 REM *****	31	200 REM *****
72	210 REM *****	22	210 REM *****	32	210 REM *****
73	220 REM *****	23	220 REM *****	33	220 REM *****
74	230 REM *****	24	230 REM *****	34	230 REM *****
75	240 REM *****	25	240 REM *****	35	240 REM *****
76	250 REM *****	26	250 REM *****	36	250 REM *****
77	260 REM *****	27	260 REM *****	37	260 REM *****
78	270 REM *****	28	270 REM *****	38	270 REM *****
79	280 REM *****	29	280 REM *****	39	280 REM *****
80	290 REM *****	30	290 REM *****	40	290 REM *****
81	300 REM *****	31	300 REM *****	41	300 REM *****
82	310 REM *****	32	310 REM *****	42	310 REM *****
83	320 REM *****	33	320 REM *****	43	320 REM *****
84	330 REM *****	34	330 REM *****	44	330 REM *****
85	340 REM *****	35	340 REM *****	45	340 REM *****
86	350 REM *****	36	350 REM *****	46	350 REM *****
87	360 REM *****	37	360 REM *****	47	360 REM *****
88	370 REM *****	38	370 REM *****	48	370 REM *****
89	380 REM *****	39	380 REM *****	49	380 REM *****
90	390 REM *****	40	390 REM *****	50	390 REM *****
91	400 REM *****	41	400 REM *****	51	400 REM *****
92	410 REM *****	42	410 REM *****	52	410 REM *****
93	420 REM *****	43	420 REM *****	53	420 REM *****
94	430 REM *****	44	430 REM *****	54	430 REM *****
95	440 REM *****	45	440 REM *****	55	440 REM *****
96	450 REM *****	46	450 REM *****	56	450 REM *****
97	460 REM *****	47	460 REM *****	57	460 REM *****
98	470 REM *****	48	470 REM *****	58	470 REM *****
99	480 REM *****	49	480 REM *****	59	480 REM *****
100	490 REM *****	50	490 REM *****	60	490 REM *****

PC Games

The PC is now a serious rival of the Amiga and C64 in the games machine race, and it's getting more popular all the time

By Tony Hetherington

The IBM PC (and its compatibles, that now include an enormous range of IBM PCs) is rapidly taking its place as a games machine. As you will see from the selection of titles listed below, the PC is not only a rival of the C64 and the Amiga in selection and quality, but it often has the first machine a game is released on. For example, Infocom's *Battletech* arrived for the PC in January, but we're still waiting for the other versions.

The PC obviously has an engineering following in the States, and so many of the games have American origins, but are now happening to find a niche over here. So at last PC owners can put aside their spreadsheets and databases and play a game.

The PC tends to attract strategy-based games that are ideally suited to a hard disk system. And remember, a PC hard disk costs only £200, which is only a third of the price of the Amiga counterpart. The PC is an option definitely worth considering.

based on the *Battletech* series of board games, in which giant mechs (fighting machines) slug it out with lasers, machine guns and missiles.

You play James Youngblood, a young trainee mech pilot who is faced with a desperate mission when the deadly Korda warriors attack and destroy your city. The only hope for your Lyran Commonwealth is to find the other survivors of the attack, and then smash down the secret sites of mech parts that will enable you to launch a counter-attack.

During the game you will have to outthink enemy bases, fight intense mechs in open combat, learn how to repair and patch up your mechs and parts and how to ward off the mutants who try to sabotage your mission.

Also from Infocom! Superb text adventures including *The Hitchhiker's Guide to the Galaxy* and the *Zork* series.

***Pool of Radiance* /SSI (US Gold)**

Dungeons and Dragons is the cult role-playing game, and so it was inevitable that there would be a computer game. Despite heavy competition, SSI won the rights to do it. *Pool of Radiance* was the first computer role-playing game and, after an initial learning stage, is unbeatable. Here at last is the game system you always wanted to play, with the character classes, monsters and spells that other systems copied, and all the original's characteristics of strength, intelligence and so on.

The combat system can at first seem very slow, especially when you're up against an army of orcs, but this soon gives on you as it gives you time to define and carry out your own



Diplomacy/Leisure Genius

Here's another game that's become a computer game, but this time

strategy and tactics. The between vital when your image score becomes strong enough to start looking around for threats and lightning bolts.

What I particularly like about *Pool of Radiance* are the unexpected surprises that it throws at you - just when you think you've got the top of things, and your confidence is growing, a trap, a monster or some other equally unpleasant situation puts you in your place.

The game also shows the value of a hard disk system as, after a lengthy installation process, you can play and save the game without ever changing a disk.

Also from SSI - *Quester II, Star Command* and *Stellar Crusade*.



***Battletech* /Infocom (Activision)**

Without doubt, *Battletech* is one of the best strategy, combat and role-playing games you will ever play on computer, and marks Infocom's transition from text-only adventures to full role-playing games. The game is

your aim is to lie, cheat and deceive your opponents. You control one of the major powers in Europe, from 1870, and must use your limited armies and fleets to take control of the continent's supply centers and complete your domination of the world.

Unfortunately, as other humans or computer players have nearly the same aims and ambitions. The only way to succeed is to back up your military might with devious schemes, and the talent to know when to strike your allies in the back, just before they plan to do it to you.

You only have two moves a year, so you can't afford to make any mistakes in a game with secret orders and cumbersome movement, credit and compromise. It's the odds that make the game so intriguing: seven players (at least one computer-controlled) and only one winner, but you can't see as your own. It's then that forces you into alliances with your enemies and to double cross your friends.

The Games - Summer Edition - Epyx (US Gold)



Eight events form the base in the Games' series of games that play just as well on the PC, as they do on other machines. This time it's the turn of the summer Olympic events that weren't covered in Summer Games I and II, including diving 400 metres hurdles (254 metres) and 100 metres, velodrome cycling (also means it's indoor), basketball (three goals), archery and the two tough gymnastic events - the rings and the parallel bars.

Up to eight players can compete for the gold, silver and bronze medals in the latest of a series that seems to have no end. The latest version, which was released in the winter, boasts improved 3D graphics to enhance the already remarkable gameplay.

Also from Epyx - California Games, World Games, Hunter games, etc.



NFL Challenge - Nor Corporation

A must for fans of football American-style. This simulated game is perhaps the best simulation of NFL. Feasible to date. The program contains actual statistics for all 28 NFL teams, and a selection of plays that leaves other games on the bench.

NFL Challenge puts you firmly in place as Head Coach and gives you the chance to call all the plays, but leaves the players to carry them out. This game gives the chance to feel the glory when long passes are caught for touchdowns, and the misery when the ball is turned over by a fumble.

You can play against either a human or computer opponent, and control all team selection and substitutions either by tactical reasons or to fill gaps left by injured players.

Whatever the ball, the players line up and carry out the plays decided by their coaches, but only in simulations (On and X) on the screen.

Also from Nor - Update disks to keep the teams state up to date.

Battle Chess/Interplay (Electronic Arts)

Take a classic, old game like chess (and they don't come much older) add some pretty graphics and animation and you've got a winner. It sounds odd, but that's exactly what happened when Interplay produced Battle Chess, a software classic.

It's a program well known to Amiga users, but is also available for the PC. It plays an average game of chess, but when one piece takes another they fight for the space with some surprising results. For example, the rooks turn into giant steam monsters that gunned their opponents while the Queen has an impressive array of magical powers, not to mention a wonderful, flame wall!

Times of Lore/Origin (Microprose)

This was one of the better attempts to bring role-playing games to the masses through combining the depth and scope of a role-playing world with the speed and reactions of arcade games.

You begin the game in a tavern knowing nothing of what lies ahead, but soon you become immersed in a quest to save the kingdom from hordes of barbarians attacking from the south and east moving from the north. All your actions are controlled by joystick movement, some windows and choosing between phrases to control conversations - in this game you must talk to people as well as killing monsters.

Your first task takes you on a raid to retrieve a magic item stolen by the ones in which you must fight your way through their guards and creep up on the camp one before striking. It's the need to organize your actions and the enemies you face that sets this game apart from all the other so-called arcade adventures.

Also from Origin Systems - Ultima II



Joan of Arc/Chip (US Gold)

Joan of Arc is a test for strategy gamers who crave power, as it crowns you king of France. Your only problem is that English forces and their treacherous allies occupy half of France. Your job is to lead them out through skill on the battlefield, strategy, diplomatic skills and rapid reflexes.

At your disposal you have generals (your best is Joan) to take charge of your armies, spies to infiltrate and assassinate your enemies, politicians to talk to and deal with your adversaries, provisions to tax to raise the money to build armies, and missionaries to keep order and pacify traitors.

The game uses action sequences to determine the outcome of battles and sieges. In these battles you use mouse, joystick or keyboard control to move soldiers, pour boiling oil on invaders, challenge soldiers to duels and lead a cavalry charge on a battlefield.

Joan of Arc features some stunning graphics, a novel setting, a good strategy base and playable events sequences that directly affect the course of the game where money, power, subterfuge, military might and diplomacy are of equal importance.

Also from US Gold = *Heroes of the Lance* and *PC Gold Wars (including Lander Board)*



Wasteland/Interplay (Electronic Arts)

This was the surprise of 1988 when it appeared on the C64 in its 8 disk sales format. Set in a post holocaust environment, you must pick what's left of civilization as a Ranger and battle with mutants and evilmen in a very unfriendly world.

Part of the fun of this disk role-playing game is to build up a party armed with a variety of weapons (anything you can find) and drive

battle tactics to use their strengths and protect their weaknesses. However, that's not all: there are also puzzles to solve, people to kill and monster (for what's left of it) to save.



Apart from saving the world and taking out anything or anybody that gets in your way, you must organize your gang headquarters to recruit kidnapped rangers, find lost children and rid farmers of the mutants that plague them. If you do all this you may be rewarded with some food, weapons or even a new party member.

The game reflects *Bard's Tale* style of role-playing, but it's not a light mutant for a change, instead of the usual selection of area and combat.

Also from Interplay = *Bard's Tale I and II*

F19/Microprose

Flight simulators have a special significance for PC owners, as among the Microsoft Simulation because the recognized way of testing whether a PC compatible was actually compatible.

Even in terms of PC flight simulators this is something special, as it gives you the chance to fly a fighter the US airforce won't even talk about. It makes you wonder whether the Russian has a PC?

The Stealth Fighter can be launched from other land bases or aircraft carriers and fly missions around America's domestic waters. That is, Libya, the Persian Gulf, the North Cape and Central Europe. Apart from the incredible attention to detail and the joys of air and cockpit you are given to help you fly the plane, the game gives you a choice of screen views that includes the usual cockpit view, as well as Top/Var, which displays you and the target on the screen. SimView lets you see the plane and the flight path, but without the gauges and TrimCam that magnifies targets for a better view of the action.

There's a choice of promotions and medals for those who survive the mission, so what more could you want?

Also from Microprose = *Gunship*, *Shred Service*, *Alabama Ranger* and *Power*



Hardware requirements

The Commodore range of PCs is just one of many so called IBM compatibles. They're compatible because IBM set the standard. However, it's a very narrow standard as there are two different disk rates, three different processors, three different standard memory configurations and four types of graphics display! The result of all that is that most games have a label somewhere on them that says something like this:

(IBM, PC, XT, AT, Compaq, Sandy 486 series 3860-4860 512K, Sep

port CGA, EGA, VGA, Hercules graphics. Supports hard disk, required DOS 2.0 or higher Japanese supported)

When buying a PC it's important to check that it works with your system, because some won't. Check that you have the required memory and run off any necessary resident programs such as SetDisk. Check also that your graphics card is brand X if it isn't it won't be supported, and finally take a careful look at whether a hard disk or even floppy drives are supported or required.

Sprite Library



This month we're going to run circles around the sprites. The CIRCLES sprites can be used as individual sequences. For example, the growing circle animation of run backgrounds will create the effect of a pebble being dropped in water. Alternatively, you can overlay one sprite over the other, the two data of overlaid will produce a clock effect. Another useful effect is to expand the range of the sprite, which will give the illusion of ID.

Table (Circle - Hires)

HEX	DECIMAL	DESCRIPTION
AB - A4	165 - 168	Growing circle
A4 - AA	168 - 170	Rotate through Y axis
AA - BD	170 - 176	Rotate through X axis
BD - BC	177 - 182	Dial large hand (clockwise)
BD - CB	183 - 200	Dial small hand (clockwise)
CB - D4	201 - 212	Disappearing Pu
D4 - D7	213 - 215	Turning wooden rail
D8 - D9	216 - 217	Turning spiked wheel
DA	218	Globe
DB	219	Yang Yang
DC - DF	220 - 223	Turning arrow

Getting it all in

Type in the basic loader as published, and SAVE IT - DON'T RUN IT, or a small self-destructor. Before running the loader program, you'll need to reset the computer and type directly the following - POKE16,0 : POKE144,0 : POKE16,0,0 : NEW and press

This month, Mike Benn shows how to run rings round sprites

This will track the computer into believing that the Basic now starts at \$2000 instead of \$4000. Load in the Basic loader and run it, if ever free, the program will automatically save itself as a block of data. If you intend that data in the future, remember to add a 1 after the device number. The data is saved in the following locations \$2000-\$7FFF.

The sprite run from 160 to 234 is a compromise to avoid the area \$2000, traditionally set aside for re-defined character graphics, and to avoid the need for typing in line after line of data.

If only one or two sprites are required then use the facilities - Sprite

block No. -160 * 40 + 160 - the data line number at which data sprite blocks data starts. Remember to type in the following three lines of data, and store the variable \$1 to the number of data lines you have in your finished program, line 1.

The small Basic program CIRCLES.DISP.LAY will variously animate the sprites in both non-expanded and expanded forms on the screen simultaneously. To hold on any sprite, enter the case number for Start and End. Any Sprite Editor program will enable you to change and adapt the individual sprites to your own requirements.



LISTINGS

[illegible]

LISTINGS

[illegible]

**GAMES
UPDATE**

[illegible]

Journalist William J. Bennett has said, "I would think [most] people would agree that the only way to get the best of the world is to get the best of the United States first." In the past, the United States has been the best of the world. It has been the most powerful, the most innovative, the most successful, and the most just of nations. It has been the most generous, the most caring, and the most compassionate of nations. It has been the most honest, the most open, and the most free of nations. It has been the most beautiful, the most peaceful, and the most prosperous of nations. It has been the most loving, the most kind, and the most forgiving of nations. It has been the most brave, the most courageous, and the most determined of nations. It has been the most wise, the most thoughtful, and the most intelligent of nations. It has been the most humble, the most modest, and the most unassuming of nations. It has been the most generous, the most caring, and the most compassionate of nations. It has been the most honest, the most open, and the most free of nations. It has been the most beautiful, the most peaceful, and the most prosperous of nations. It has been the most loving, the most kind, and the most forgiving of nations. It has been the most brave, the most courageous, and the most determined of nations. It has been the most wise, the most thoughtful, and the most intelligent of nations. It has been the most humble, the most modest, and the most unassuming of nations.

I argued, if I couldn't take a walk to the park, I'd be better off staying home. I'd rather stay home than go to a party where I'd be bored. But the doctor said, "You're overreacting. You're not sick. You're just tired. You need to go to the park. It'll make you feel better." I decided that I was a little bit sick, but I'd go to the park anyway. And when I got there, I was fine.

Keywords: children; divorce; depression; self-esteem; social support
The authors thank Susan Marshall and T.J. Paul for their advice during preparation.

LAS VEGAS CASINO

Despite the title, Las Vegas Casino offers none of the excitement of the real thing. Here is your chance to lose pretend Monopoly money at four different casinos. Black jack, baccarat and craps.

Starting with only 120 pounds, can you break the bank? Frailly, who says? Certainly, there is no lightness way it can be done in real life and just to make doubly sure, the exercise sheets on a trail does not use anyone rules.

Take Black Jack for example. Now if you are a good gambler, this is the game that you want to make most money at. For every pound that you invest, you should be able to get 99 pence back. The way to win (or not lose as much), is to know when to make favourable bets and when to stick and twist. This can be done because you know that the dealer must beat an 18 and stick on 17.

In this version, the program starts as soon as it has got a better hand than yours, so all thoughts of tactical betting go out of the window. OK, so you might get lucky in the short term, but the odds over a period of time are totally stacked against you.

Place graphics and controls, especially on the roulette wheel, only before the most blurring moments of the game. When well-programmed authors realize that gambling when there is nothing at stake is utterly pointless? Save your money or, if you must blow three grand, put it on the 3:28 at Aintree. At least there'll be some momentary excitement, even if your horse does come in last.

100

Falls: Lee Vigue, Owner, Supplier, Zepherus Games, 38
Chalmers Road, Arcadia, New South Wales 2122, A.S.
Phone: 12 802



GAMES UPDATE

Para Assault Course

Years ago, I can remember wiggling my joystick like mad as I tried to survive a grueling 1500 meters run in a Breakline game. Surprised, surprise, the same idea has resurfaced again. Oh to be sure, the game looks superficially different. The background here is a Parashute regiment assault course rather than an athletic stadium, but at the end of the day, joystick wiggling is what it is all about.

There are four different courses for you to attempt, each containing a variety of obstacles. There are walls to be climbed, and ponds of water to be leapt. There are rings and rope swings, ladders and death slides and all the time, you are competing against the clock.

At a time when the news has just broken about nuclear fusion, there is a new energy factor at work here. Run into the wall from a distance of one yard and bounce back three!

Each course can be previewed and practised before you attempt the real thing, although telling you for falling off the death slide does seem a little harsh. Fair enough when you're trying the course proper, but surely not to practise mode.

I didn't enjoy this game at all. Yes, I know it's a version of certain full price games currently available, and that will obviously tempt people to buy it, but it seems to be lacking one vital ingredient - fun. And that after all, is what games are all about.



© Copyright 1989 by the publisher, Breakline, Inc. All rights reserved. This game is a registered trademark of Breakline, Inc. All other trademarks are the property of their respective owners.

Touchline
Title: Para Assault Course
Developer: Breakline, Inc.
Publisher: Breakline, Inc.
Price: \$29.95

Head for Home

We present an update for this popular program

By Kirk McMillan

Head for Home was written using the technique of giving ML Data onto the end of a basic program. A few zero bytes separate each section and prevent the ML from being displayed with Basic's LIST command.

There are two very good reasons for doing this. Firstly, it, saves considerable room. Head for Home would not have fitted onto the C-16's memory otherwise, and it's faster running, with no waiting for FOR NEXT loops to poke data into memory.

However, there is one disadvantage. Basic programs can't be edited or stored. This changes the number of bytes used, and causes the ML routines to be at a different address to that intended. The only exception is where the change doesn't affect the length of a program - as in changing DLOAD to LOAD. Both commands are tokenised and are not byte.

Unfortunately, Head for Home's machine language wasn't spotted by us at the magazine and was published as two basic programs (14c March, 1984).

Now comes the task of adding the ML to both programs.

After typing in the basic sections, you must check that no errors appear as Basic have extended the programs to where the ML is to go. Load the first program as usual, in direct mode, onto

PRINTPEEK (40) * 256 * PEEK (40)

The answer must be less than 4096. Likewise, the result for the second program should be less than 9600. If not receive anything from the programs that is not to the listing.

The ML data could be entered

directly into the monitor, but that's a lot of typing required with no means of error checking. DATA ENTRY may help a little here by speeding up input and storing you to memory.

Type in and run the program - it will first ask you for a starting address. Enter the last number from the listing e.g. to begin with it would be 1064. You will need to keep track of where you're up to if you type in the data over several listings.

When a line is typed in the program will generate a checksum which you can check against the listing. Press RETURN only if it matches, any other key allows you to re-type before.

To Exit from DATA ENTRY - press ESC twice and then enter the amount to save your work.

FOR ML: use S "name", 1, 1064, 1064

and ML: S "name", 1, 256, 300

To continue from on, LOAD "name", 1, then NEW, DLOAD "Data Entry"

and RUN (if using tape substitute "1" for "1" and LOAD for DLOAD)

If you lose track of where you are, or for any reason want to check what has been entered, then simply use the "M" command from within Teletext. The line addresses and each row of bytes will correspond 1 to 1 with the listing. You should use the "T" command to fill memory with a known value to update where data has or hasn't been entered.

Over both ML programs are finished, (ML1 should be 3 disk blocks, and ML2 10 blocks), the method for merging these with the Basic programs is as follows:

- (1) DLOAD "Head for Home" in the normal manner
- (2) RENAME "Head for Home" TO "HFM.sit"
- (3) LOAD "ml1", 1, 1
- (4) SAVE "Head for Home"
- (5) DLOAD "L. BASIC"
- (6) RENAME "1 basic" to "1 basic old"
- (7) LOAD "ml2", 1, 1
- (8) SAVE "1 basic"

DATA ENTRY

```
10 DEFEND
20 PRINT "DATA ENTRY FOR ML"
30 IF CLIP PRINT "CHECKSUM STARTED"
40 I = 0
50 GETVARS IN=IN+1:IN=IN+1:IN=IN+1
60 IF IN=IN+1 ELSE 100
70 IF IN=IN+1 ELSE 100
80 IF IN=IN+1 ELSE 100
90 IF IN=IN+1 ELSE 100
100 IF IN=IN+1 ELSE 100
110 IF IN=IN+1 ELSE 100
120 IF IN=IN+1 ELSE 100
130 IF IN=IN+1 ELSE 100
140 IF IN=IN+1 ELSE 100
150 IF IN=IN+1 ELSE 100
160 IF IN=IN+1 ELSE 100
170 IF IN=IN+1 ELSE 100
180 IF IN=IN+1 ELSE 100
190 IF IN=IN+1 ELSE 100
200 IF IN=IN+1 ELSE 100
210 IF IN=IN+1 ELSE 100
220 IF IN=IN+1 ELSE 100
230 IF IN=IN+1 ELSE 100
240 IF IN=IN+1 ELSE 100
250 IF IN=IN+1 ELSE 100
260 IF IN=IN+1 ELSE 100
270 IF IN=IN+1 ELSE 100
280 IF IN=IN+1 ELSE 100
290 IF IN=IN+1 ELSE 100
300 IF IN=IN+1 ELSE 100
310 IF IN=IN+1 ELSE 100
320 IF IN=IN+1 ELSE 100
330 IF IN=IN+1 ELSE 100
340 IF IN=IN+1 ELSE 100
350 IF IN=IN+1 ELSE 100
360 IF IN=IN+1 ELSE 100
370 IF IN=IN+1 ELSE 100
380 IF IN=IN+1 ELSE 100
390 IF IN=IN+1 ELSE 100
400 IF IN=IN+1 ELSE 100
410 IF IN=IN+1 ELSE 100
420 IF IN=IN+1 ELSE 100
430 IF IN=IN+1 ELSE 100
440 IF IN=IN+1 ELSE 100
450 IF IN=IN+1 ELSE 100
460 IF IN=IN+1 ELSE 100
470 IF IN=IN+1 ELSE 100
480 IF IN=IN+1 ELSE 100
490 IF IN=IN+1 ELSE 100
500 IF IN=IN+1 ELSE 100
510 IF IN=IN+1 ELSE 100
520 IF IN=IN+1 ELSE 100
530 IF IN=IN+1 ELSE 100
540 IF IN=IN+1 ELSE 100
550 IF IN=IN+1 ELSE 100
560 IF IN=IN+1 ELSE 100
570 IF IN=IN+1 ELSE 100
580 IF IN=IN+1 ELSE 100
590 IF IN=IN+1 ELSE 100
600 IF IN=IN+1 ELSE 100
610 IF IN=IN+1 ELSE 100
620 IF IN=IN+1 ELSE 100
630 IF IN=IN+1 ELSE 100
640 IF IN=IN+1 ELSE 100
650 IF IN=IN+1 ELSE 100
660 IF IN=IN+1 ELSE 100
670 IF IN=IN+1 ELSE 100
680 IF IN=IN+1 ELSE 100
690 IF IN=IN+1 ELSE 100
700 IF IN=IN+1 ELSE 100
710 IF IN=IN+1 ELSE 100
720 IF IN=IN+1 ELSE 100
730 IF IN=IN+1 ELSE 100
740 IF IN=IN+1 ELSE 100
750 IF IN=IN+1 ELSE 100
760 IF IN=IN+1 ELSE 100
770 IF IN=IN+1 ELSE 100
780 IF IN=IN+1 ELSE 100
790 IF IN=IN+1 ELSE 100
800 IF IN=IN+1 ELSE 100
810 IF IN=IN+1 ELSE 100
820 IF IN=IN+1 ELSE 100
830 IF IN=IN+1 ELSE 100
840 IF IN=IN+1 ELSE 100
850 IF IN=IN+1 ELSE 100
860 IF IN=IN+1 ELSE 100
870 IF IN=IN+1 ELSE 100
880 IF IN=IN+1 ELSE 100
890 IF IN=IN+1 ELSE 100
900 IF IN=IN+1 ELSE 100
910 IF IN=IN+1 ELSE 100
920 IF IN=IN+1 ELSE 100
930 IF IN=IN+1 ELSE 100
940 IF IN=IN+1 ELSE 100
950 IF IN=IN+1 ELSE 100
960 IF IN=IN+1 ELSE 100
970 IF IN=IN+1 ELSE 100
980 IF IN=IN+1 ELSE 100
990 IF IN=IN+1 ELSE 100
1000 IF IN=IN+1 ELSE 100
1010 IF IN=IN+1 ELSE 100
1020 IF IN=IN+1 ELSE 100
1030 IF IN=IN+1 ELSE 100
1040 IF IN=IN+1 ELSE 100
1050 IF IN=IN+1 ELSE 100
1060 IF IN=IN+1 ELSE 100
1070 IF IN=IN+1 ELSE 100
1080 IF IN=IN+1 ELSE 100
1090 IF IN=IN+1 ELSE 100
1100 IF IN=IN+1 ELSE 100
1110 IF IN=IN+1 ELSE 100
1120 IF IN=IN+1 ELSE 100
1130 IF IN=IN+1 ELSE 100
1140 IF IN=IN+1 ELSE 100
1150 IF IN=IN+1 ELSE 100
1160 IF IN=IN+1 ELSE 100
1170 IF IN=IN+1 ELSE 100
1180 IF IN=IN+1 ELSE 100
1190 IF IN=IN+1 ELSE 100
1200 IF IN=IN+1 ELSE 100
1210 IF IN=IN+1 ELSE 100
1220 IF IN=IN+1 ELSE 100
1230 IF IN=IN+1 ELSE 100
1240 IF IN=IN+1 ELSE 100
1250 IF IN=IN+1 ELSE 100
1260 IF IN=IN+1 ELSE 100
1270 IF IN=IN+1 ELSE 100
1280 IF IN=IN+1 ELSE 100
1290 IF IN=IN+1 ELSE 100
1300 IF IN=IN+1 ELSE 100
1310 IF IN=IN+1 ELSE 100
1320 IF IN=IN+1 ELSE 100
1330 IF IN=IN+1 ELSE 100
1340 IF IN=IN+1 ELSE 100
1350 IF IN=IN+1 ELSE 100
1360 IF IN=IN+1 ELSE 100
1370 IF IN=IN+1 ELSE 100
1380 IF IN=IN+1 ELSE 100
1390 IF IN=IN+1 ELSE 100
1400 IF IN=IN+1 ELSE 100
1410 IF IN=IN+1 ELSE 100
1420 IF IN=IN+1 ELSE 100
1430 IF IN=IN+1 ELSE 100
1440 IF IN=IN+1 ELSE 100
1450 IF IN=IN+1 ELSE 100
1460 IF IN=IN+1 ELSE 100
1470 IF IN=IN+1 ELSE 100
1480 IF IN=IN+1 ELSE 100
1490 IF IN=IN+1 ELSE 100
1500 IF IN=IN+1 ELSE 100
1510 IF IN=IN+1 ELSE 100
1520 IF IN=IN+1 ELSE 100
1530 IF IN=IN+1 ELSE 100
1540 IF IN=IN+1 ELSE 100
1550 IF IN=IN+1 ELSE 100
1560 IF IN=IN+1 ELSE 100
1570 IF IN=IN+1 ELSE 100
1580 IF IN=IN+1 ELSE 100
1590 IF IN=IN+1 ELSE 100
1600 IF IN=IN+1 ELSE 100
1610 IF IN=IN+1 ELSE 100
1620 IF IN=IN+1 ELSE 100
1630 IF IN=IN+1 ELSE 100
1640 IF IN=IN+1 ELSE 100
1650 IF IN=IN+1 ELSE 100
1660 IF IN=IN+1 ELSE 100
1670 IF IN=IN+1 ELSE 100
1680 IF IN=IN+1 ELSE 100
1690 IF IN=IN+1 ELSE 100
1700 IF IN=IN+1 ELSE 100
1710 IF IN=IN+1 ELSE 100
1720 IF IN=IN+1 ELSE 100
1730 IF IN=IN+1 ELSE 100
1740 IF IN=IN+1 ELSE 100
1750 IF IN=IN+1 ELSE 100
1760 IF IN=IN+1 ELSE 100
1770 IF IN=IN+1 ELSE 100
1780 IF IN=IN+1 ELSE 100
1790 IF IN=IN+1 ELSE 100
1800 IF IN=IN+1 ELSE 100
1810 IF IN=IN+1 ELSE 100
1820 IF IN=IN+1 ELSE 100
1830 IF IN=IN+1 ELSE 100
1840 IF IN=IN+1 ELSE 100
1850 IF IN=IN+1 ELSE 100
1860 IF IN=IN+1 ELSE 100
1870 IF IN=IN+1 ELSE 100
1880 IF IN=IN+1 ELSE 100
1890 IF IN=IN+1 ELSE 100
1900 IF IN=IN+1 ELSE 100
1910 IF IN=IN+1 ELSE 100
1920 IF IN=IN+1 ELSE 100
1930 IF IN=IN+1 ELSE 100
1940 IF IN=IN+1 ELSE 100
1950 IF IN=IN+1 ELSE 100
1960 IF IN=IN+1 ELSE 100
1970 IF IN=IN+1 ELSE 100
1980 IF IN=IN+1 ELSE 100
1990 IF IN=IN+1 ELSE 100
2000 IF IN=IN+1 ELSE 100
2010 IF IN=IN+1 ELSE 100
2020 IF IN=IN+1 ELSE 100
2030 IF IN=IN+1 ELSE 100
2040 IF IN=IN+1 ELSE 100
2050 IF IN=IN+1 ELSE 100
2060 IF IN=IN+1 ELSE 100
2070 IF IN=IN+1 ELSE 100
2080 IF IN=IN+1 ELSE 100
2090 IF IN=IN+1 ELSE 100
2100 IF IN=IN+1 ELSE 100
2110 IF IN=IN+1 ELSE 100
2120 IF IN=IN+1 ELSE 100
2130 IF IN=IN+1 ELSE 100
2140 IF IN=IN+1 ELSE 100
2150 IF IN=IN+1 ELSE 100
2160 IF IN=IN+1 ELSE 100
2170 IF IN=IN+1 ELSE 100
2180 IF IN=IN+1 ELSE 100
2190 IF IN=IN+1 ELSE 100
2200 IF IN=IN+1 ELSE 100
2210 IF IN=IN+1 ELSE 100
2220 IF IN=IN+1 ELSE 100
2230 IF IN=IN+1 ELSE 100
2240 IF IN=IN+1 ELSE 100
2250 IF IN=IN+1 ELSE 100
2260 IF IN=IN+1 ELSE 100
2270 IF IN=IN+1 ELSE 100
2280 IF IN=IN+1 ELSE 100
2290 IF IN=IN+1 ELSE 100
2300 IF IN=IN+1 ELSE 100
2310 IF IN=IN+1 ELSE 100
2320 IF IN=IN+1 ELSE 100
2330 IF IN=IN+1 ELSE 100
2340 IF IN=IN+1 ELSE 100
2350 IF IN=IN+1 ELSE 100
2360 IF IN=IN+1 ELSE 100
2370 IF IN=IN+1 ELSE 100
2380 IF IN=IN+1 ELSE 100
2390 IF IN=IN+1 ELSE 100
2400 IF IN=IN+1 ELSE 100
2410 IF IN=IN+1 ELSE 100
2420 IF IN=IN+1 ELSE 100
2430 IF IN=IN+1 ELSE 100
2440 IF IN=IN+1 ELSE 100
2450 IF IN=IN+1 ELSE 100
2460 IF IN=IN+1 ELSE 100
2470 IF IN=IN+1 ELSE 100
2480 IF IN=IN+1 ELSE 100
2490 IF IN=IN+1 ELSE 100
2500 IF IN=IN+1 ELSE 100
2510 IF IN=IN+1 ELSE 100
2520 IF IN=IN+1 ELSE 100
2530 IF IN=IN+1 ELSE 100
2540 IF IN=IN+1 ELSE 100
2550 IF IN=IN+1 ELSE 100
2560 IF IN=IN+1 ELSE 100
2570 IF IN=IN+1 ELSE 100
2580 IF IN=IN+1 ELSE 100
2590 IF IN=IN+1 ELSE 100
2600 IF IN=IN+1 ELSE 100
2610 IF IN=IN+1 ELSE 100
2620 IF IN=IN+1 ELSE 100
2630 IF IN=IN+1 ELSE 100
2640 IF IN=IN+1 ELSE 100
2650 IF IN=IN+1 ELSE 100
2660 IF IN=IN+1 ELSE 100
2670 IF IN=IN+1 ELSE 100
2680 IF IN=IN+1 ELSE 100
2690 IF IN=IN+1 ELSE 100
2700 IF IN=IN+1 ELSE 100
2710 IF IN=IN+1 ELSE 100
2720 IF IN=IN+1 ELSE 100
2730 IF IN=IN+1 ELSE 100
2740 IF IN=IN+1 ELSE 100
2750 IF IN=IN+1 ELSE 100
2760 IF IN=IN+1 ELSE 100
2770 IF IN=IN+1 ELSE 100
2780 IF IN=IN+1 ELSE 100
2790 IF IN=IN+1 ELSE 100
2800 IF IN=IN+1 ELSE 100
2810 IF IN=IN+1 ELSE 100
2820 IF IN=IN+1 ELSE 100
2830 IF IN=IN+1 ELSE 100
2840 IF IN=IN+1 ELSE 100
2850 IF IN=IN+1 ELSE 100
2860 IF IN=IN+1 ELSE 100
2870 IF IN=IN+1 ELSE 100
2880 IF IN=IN+1 ELSE 100
2890 IF IN=IN+1 ELSE 100
2900 IF IN=IN+1 ELSE 100
2910 IF IN=IN+1 ELSE 100
2920 IF IN=IN+1 ELSE 100
2930 IF IN=IN+1 ELSE 100
2940 IF IN=IN+1 ELSE 100
2950 IF IN=IN+1 ELSE 100
2960 IF IN=IN+1 ELSE 100
2970 IF IN=IN+1 ELSE 100
2980 IF IN=IN+1 ELSE 100
2990 IF IN=IN+1 ELSE 100
3000 IF IN=IN+1 ELSE 100
3010 IF IN=IN+1 ELSE 100
3020 IF IN=IN+1 ELSE 100
3030 IF IN=IN+1 ELSE 100
3040 IF IN=IN+1 ELSE 100
3050 IF IN=IN+1 ELSE 100
3060 IF IN=IN+1 ELSE 100
3070 IF IN=IN+1 ELSE 100
3080 IF IN=IN+1 ELSE 100
3090 IF IN=IN+1 ELSE 100
3100 IF IN=IN+1 ELSE 100
3110 IF IN=IN+1 ELSE 100
3120 IF IN=IN+1 ELSE 100
3130 IF IN=IN+1 ELSE 100
3140 IF IN=IN+1 ELSE 100
3150 IF IN=IN+1 ELSE 100
3160 IF IN=IN+1 ELSE 100
3170 IF IN=IN+1 ELSE 100
3180 IF IN=IN+1 ELSE 100
3190 IF IN=IN+1 ELSE 100
3200 IF IN=IN+1 ELSE 100
3210 IF IN=IN+1 ELSE 100
3220 IF IN=IN+1 ELSE 100
3230 IF IN=IN+1 ELSE 100
3240 IF IN=IN+1 ELSE 100
3250 IF IN=IN+1 ELSE 100
3260 IF IN=IN+1 ELSE 100
3270 IF IN=IN+1 ELSE 100
3280 IF IN=IN+1 ELSE 100
3290 IF IN=IN+1 ELSE 100
3300 IF IN=IN+1 ELSE 100
3310 IF IN=IN+1 ELSE 100
3320 IF IN=IN+1 ELSE 100
3330 IF IN=IN+1 ELSE 100
3340 IF IN=IN+1 ELSE 100
3350 IF IN=IN+1 ELSE 100
3360 IF IN=IN+1 ELSE 100
3370 IF IN=IN+1 ELSE 100
3380 IF IN=IN+1 ELSE 100
3390 IF IN=IN+1 ELSE 100
3400 IF IN=IN+1 ELSE 100
3410 IF IN=IN+1 ELSE 100
3420 IF IN=IN+1 ELSE 100
3430 IF IN=IN+1 ELSE 100
3440 IF IN=IN+1 ELSE 100
3450 IF IN=IN+1 ELSE 100
3460 IF IN=IN+1 ELSE 100
3470 IF IN=IN+1 ELSE 100
3480 IF IN=IN+1 ELSE 100
3490 IF IN=IN+1 ELSE 100
3500 IF IN=IN+1 ELSE 100
3510 IF IN=IN+1 ELSE 100
3520 IF IN=IN+1 ELSE 100
3530 IF IN=IN+1 ELSE 100
3540 IF IN=IN+1 ELSE 100
3550 IF IN=IN+1 ELSE 100
3560 IF IN=IN+1 ELSE 100
3570 IF IN=IN+1 ELSE 100
3580 IF IN=IN+1 ELSE 100
3590 IF IN=IN+1 ELSE 100
3600 IF IN=IN+1 ELSE 100
3610 IF IN=IN+1 ELSE 100
3620 IF IN=IN+1 ELSE 100
3630 IF IN=IN+1 ELSE 100
3640 IF IN=IN+1 ELSE 100
3650 IF IN=IN+1 ELSE 100
3660 IF IN=IN+1 ELSE 100
3670 IF IN=IN+1 ELSE 100
3680 IF IN=IN+1 ELSE 100
3690 IF IN=IN+1 ELSE 100
3700 IF IN=IN+1 ELSE 100
3710 IF IN=IN+1 ELSE 100
3720 IF IN=IN+1 ELSE 100
3730 IF IN=IN+1 ELSE 100
3740 IF IN=IN+1 ELSE 100
3750 IF IN=IN+1 ELSE 100
3760 IF IN=IN+1 ELSE 100
3770 IF IN=IN+1 ELSE 100
3780 IF IN=IN+1 ELSE 100
3790 IF IN=IN+1 ELSE 100
3800 IF IN=IN+1 ELSE 100
3810 IF IN=IN+1 ELSE 100
3820 IF IN=IN+1 ELSE 100
3830 IF IN=IN+1 ELSE 100
3840 IF IN=IN+1 ELSE 100
3850 IF IN=IN+1 ELSE 100
3860 IF IN=IN+1 ELSE 100
3870 IF IN=IN+1 ELSE 100
3880 IF IN=IN+1 ELSE 100
3890 IF IN=IN+1 ELSE 100
3900 IF IN=IN+1 ELSE 100
3910 IF IN=IN+1 ELSE 100
3920 IF IN=IN+1 ELSE 100
3930 IF IN=IN+1 ELSE 100
3940 IF IN=IN+1 ELSE 100
3950 IF IN=IN+1 ELSE 100
3960 IF IN=IN+1 ELSE 100
3970 IF IN=IN+1 ELSE 100
3980 IF IN=IN+1 ELSE 100
3990 IF IN=IN+1 ELSE 100
4000 IF IN=IN+1 ELSE 100
4010 IF IN=IN+1 ELSE 100
4020 IF IN=IN+1 ELSE 100
4030 IF IN=IN+1 ELSE 100
4040 IF IN=IN+1 ELSE 100
4050 IF IN=IN+1 ELSE 100
4060 IF IN=IN+1 ELSE 100
4070 IF IN=IN+1 ELSE 100
4080 IF IN=IN+1 ELSE 100
4090 IF IN=IN+1 ELSE 100
4100 IF IN=IN+1 ELSE 100
4110 IF IN=IN+1 ELSE 100
4120 IF IN=IN+1 ELSE 100
4130 IF IN=IN+1 ELSE 100
4140 IF IN=IN+1 ELSE 100
4150 IF IN=IN+1 ELSE 100
4160 IF IN=IN+1 ELSE 100
4170 IF IN=IN+1 ELSE 100
4180 IF IN=IN+1 ELSE 100
4190 IF IN=IN+1 ELSE 100
4200 IF IN=IN+1 ELSE 100
4210 IF IN=IN+1 ELSE 100
4220 IF IN=IN+1 ELSE 100
4230 IF IN=IN+1 ELSE 100
4240 IF IN=IN+1 ELSE 100
4250 IF IN=IN+1 ELSE 100
4260 IF IN=IN+1 ELSE 100
4270 IF IN=IN+1 ELSE 100
4280 IF IN=IN+1 ELSE 100
4290 IF IN=IN+1 ELSE 100
4300 IF IN=IN+1 ELSE 100
4310 IF IN=IN+1 ELSE 100
4320 IF IN=IN+1 ELSE 100
4330 IF IN=IN+1 ELSE 100
4340 IF IN=IN+1 ELSE 100
4350 IF IN=IN+1 ELSE 100
4360 IF IN=IN+1 ELSE 100
4370 IF IN=IN+1 ELSE 100
4380 IF IN=IN+1 ELSE 100
4390 IF IN=IN+1 ELSE 100
4400 IF IN=IN+1 ELSE 100
4410 IF IN=IN+1 ELSE 100
4420 IF IN=IN+1 ELSE 100
4430 IF IN=IN+1 ELSE 100
4440 IF IN=IN+1 ELSE 100
4450 IF IN=IN+1 ELSE 100
4460 IF IN=IN+1 ELSE 100
4470 IF IN=IN+1 ELSE 100
4480 IF IN=IN+1 ELSE 100
4490 IF IN=IN+1 ELSE 100
4500 IF IN=IN+1 ELSE 100
4510 IF IN=IN+1 ELSE 100
4520 IF IN=IN+1 ELSE 100
4530 IF IN=IN+1 ELSE 100
4540 IF IN=IN+1 ELSE 100
4550 IF IN=IN+1 ELSE 100
4560 IF IN=IN+1 ELSE 100
4570 IF IN=IN+1 ELSE 100
4580 IF IN=IN+1 ELSE 100
4590 IF IN=IN+1 ELSE 100
4600 IF IN=IN+1 ELSE 100
4610 IF IN=IN+1 ELSE 100
4620 IF IN=IN+1 ELSE 100
4630 IF IN=IN+1 ELSE 100
4640 IF IN=IN+1 ELSE 100
4650 IF IN=IN+1 ELSE 100
4660 IF IN=IN+1 ELSE 100
4670 IF IN=IN+1 ELSE 100
4680 IF IN=IN+1 ELSE 100
4690 IF IN=IN+1 ELSE 100
4700 IF IN=IN+1 ELSE 100
4710 IF IN=IN+1 ELSE 100
4720 IF IN=IN+1 ELSE 100
4730 IF IN=IN+1 ELSE 100
4740 IF IN=IN+1 ELSE 100
4750 IF IN=IN+1 ELSE 100
4760 IF IN=IN+1 ELSE 100
4770 IF IN=IN+1 ELSE 100
4780 IF IN=IN+1 ELSE 100
4790 IF IN=IN+1 ELSE 100
4800 IF IN=IN+1 ELSE 100
4810 IF IN=IN+1 ELSE 100
4820 IF IN=IN+1 ELSE 100
4830 IF IN=IN+1 ELSE 100
4840 IF IN=IN+1 ELSE 100
4850 IF IN=IN+1 ELSE 100
4860 IF IN=IN+1 ELSE 100
4870 IF IN=IN+1 ELSE 100
4880 IF IN=IN+1 ELSE 100
4890 IF IN=IN+1 ELSE 100
4900 IF IN=IN+1 ELSE 100
4910 IF IN=IN+1 ELSE 100
4920 IF IN=IN+1 ELSE 100
4930 IF IN=IN+1 ELSE 100
4940 IF IN=IN+1 ELSE 100
4950 IF IN=IN+1 ELSE 100
4960 IF IN=IN+1 ELSE 100
4970 IF IN=IN+1 ELSE 100
4980 IF IN=IN+1 ELSE 100
4990 IF IN=IN+1 ELSE 100
5000 IF IN=IN+1 ELSE 100
5010 IF IN=IN+1 ELSE 100
5020 IF IN=IN+1 ELSE 100
5030 IF IN=IN+1 ELSE 100
5040 IF IN=IN+1 ELSE 100
5050 IF IN=IN+1 ELSE 100
5060 IF IN=IN+1 ELSE 100
5070 IF IN=IN+1 ELSE 100
5080 IF IN=IN+1 ELSE 100
5090 IF IN=IN+1 ELSE 100
5100 IF IN=IN+1 ELSE 100
5110 IF IN=IN+1 ELSE 100
5120 IF IN=IN+1 ELSE 100
5130 IF IN=IN+1 ELSE 100
5140 IF IN=IN+1 ELSE 100
5150 IF IN=IN+1 ELSE 100
5160 IF IN=IN+1 ELSE 100
5170 IF IN=IN+1 ELSE 100
5180 IF IN=IN+1 ELSE 100
5190 IF IN=IN+1 ELSE 100
5200 IF IN=IN+1 ELSE 100
5210 IF IN=IN+1 ELSE 100
5220 IF IN=IN+1 ELSE 100
5230 IF IN=IN+1 ELSE 100
5240 IF IN=IN+1 ELSE 100
5250 IF IN=IN+1 ELSE 100
5260 IF IN=IN+1 ELSE 100
5270 IF IN=IN+1 ELSE 100
5280 IF IN=IN+1 ELSE 100
5290 IF IN=IN+1 ELSE 100
5300 IF IN=IN+1 ELSE 100
5310 IF IN=IN+1 ELSE 100
5320 IF IN=IN+1 ELSE 100
5330 IF IN=IN+1 ELSE 100
5340 IF IN=IN+1 ELSE 100
5350 IF IN=IN+1 ELSE 100
5360 IF IN=IN+1 ELSE 100
5370 IF IN=IN+1 ELSE 100
5380 IF IN=IN+1 ELSE 100
5390 IF IN=IN+1 ELSE 100
5400 IF IN=IN+1 ELSE 100
5410 IF IN=IN+1 ELSE 100
5420 IF IN=IN+1 ELSE 100
5430 IF IN=IN+1 ELSE 100
5440 IF IN=IN+1 ELSE 100
5450 IF IN=IN+1 ELSE 100
5460 IF IN=IN+1 ELSE 100
5470 IF IN=IN+1 ELSE 100
5480 IF IN=IN+1 ELSE 100
5490 IF IN=IN+1 ELSE 100
5500 IF IN=IN+1 ELSE 100
5510 IF IN=IN+1 ELSE 100
5520 IF IN=IN+1 ELSE 100
5530 IF IN=IN+1 ELSE 100
5540 IF IN=IN+1 ELSE 100
5550 IF IN=IN+1 ELSE 100
5560 IF IN=IN+1 ELSE 100
5570 IF IN=IN+1 ELSE 100
5580 IF IN=IN+1 ELSE 100
5590 IF IN=IN+1 ELSE 100
5600 IF IN=IN+1 ELSE 100
5610 IF IN=IN+1 ELSE 100
5620 IF IN=IN+1 ELSE 100
5630 IF IN=IN+1 ELSE 100
5640 IF IN=IN+1 ELSE 100
5650 IF IN=IN+1 ELSE 100
5660 IF IN=IN+1 ELSE 100
5670 IF IN=IN+1 ELSE 100
5680 IF IN=IN+1 ELSE 100
5690 IF IN=IN+1 ELSE 100
5700 IF IN=IN+1 ELSE 100
5710 IF IN=IN+1 ELSE 100
5720 IF IN=IN+1 ELSE 100
5730 IF IN=IN+1 ELSE 100
5740 IF IN=IN+1 ELSE 100
5750 IF IN=IN+1 ELSE 100
5760 IF IN=IN+1 ELSE 100
5770 IF IN=IN+1 ELSE 100
5780 IF IN=IN+1 ELSE 100
5790 IF IN=IN+1 ELSE 100
5800 IF IN=IN+1 ELSE 100
5810 IF IN=IN+1 ELSE 100
5820 IF IN=IN+1 ELSE 100
5830 IF IN=IN+1 ELSE 100
5840 IF IN=IN+1 ELSE 100
5850 IF IN=IN+1 ELSE 100
5860 IF IN=IN+1 ELSE 100
5870 IF IN=IN+1 ELSE 100
5880 IF IN=IN+1 ELSE 100
5890 IF IN=IN+1 ELSE 100
5900 IF IN=IN+1 ELSE 100
5910 IF IN=IN+1 ELSE 100
5920 IF IN=IN+1 ELSE 100
5930 IF IN=IN+1 ELSE 100
5940 IF IN=IN+1 ELSE 100
5950 IF IN=IN+1 ELSE 100
5960 IF IN=IN+1 ELSE 100
5970 IF IN=IN+1 ELSE 100
5980 IF IN=IN+1 ELSE 100
5990 IF IN=IN+1 ELSE 100
6000 IF IN=IN+1 ELSE 100
6010 IF IN=IN+1 ELSE 100
6020 IF IN=IN+1 ELSE 100
6030 IF IN=IN+1 ELSE 100
6040 IF IN=IN+1 ELSE 100
6050 IF IN=IN+1 ELSE 100
6060 IF IN=IN+1 ELSE 100
6070 IF IN=IN+1 ELSE 100
6080 IF IN=IN+1 ELSE 100
6090 IF IN=IN+1 ELSE 100
6100 IF IN=IN+1 ELSE 100
6110 IF IN=IN+1 ELSE 100
6120 IF IN=IN+1 ELSE 100
6130 IF IN=IN+1 ELSE 100
6140 IF IN=IN+1 ELSE 100
6150 IF IN=IN+1 ELSE 100
6160 IF IN=IN+1 ELSE 100
6170 IF IN=IN+1 ELSE 100
6180 IF IN=IN+1 ELSE 100
6190 IF IN=IN+1 ELSE 100
6200 IF IN=IN+1 ELSE 100
6210 IF IN=IN+1 ELSE 100
6220 IF IN=IN+1 ELSE 100
6230 IF IN=IN+1 ELSE 100
6240 IF IN=IN+1 ELSE 100
6250 IF IN=IN+1 ELSE 100
6260 IF IN=IN+1 ELSE 100
6270 IF IN=IN+1 ELSE 100
6280 IF IN=IN+1 ELSE 100
6290 IF IN=IN+1 ELSE 100
6300 IF IN=IN+1 ELSE 100
6310 IF IN=IN+1 ELSE 100
6320 IF IN=IN+1 ELSE 100
6330 IF IN=IN+1 ELSE 100
6340 IF IN=IN+1 ELSE 100
6350 IF IN=IN+1 ELSE 100
6360 IF IN=IN+1 ELSE 100
6370 IF IN=IN+1 ELSE 100
6380 IF IN=IN+1 ELSE 100
6390 IF IN=IN+1 ELSE 100
6400 IF IN=IN+1 ELSE 100
6410 IF IN=IN+1 ELSE 100
6420 IF IN=IN+1 ELSE 100
6430 IF IN=IN+1 ELSE 100
6440 IF IN=IN+1 ELSE 100
6450 IF IN=IN+1 ELSE 100
6460 IF IN=IN+1 ELSE 100
6470 IF IN=IN+1 ELSE 100
6480 IF IN=IN+1 ELSE 100
6490 IF IN=IN+1 ELSE 100
6500 IF IN=IN+1 ELSE 100
6510 IF IN=IN+1 ELSE 100
6520 IF IN=IN+1 ELSE 100
6530 IF IN=IN+1 ELSE 100
6540 IF IN=IN+1 ELSE 100
6550 IF IN=IN+1 ELSE 100
6560 IF IN=IN+1 ELSE 100
6570 IF IN=IN+1 ELSE 100
6580 IF IN=IN+1 ELSE 100
6590 IF IN=IN+1 ELSE 100
6600 IF IN=IN+1 ELSE 100
6610 IF IN=IN+1 ELSE 100
6620 IF IN=IN+1 ELSE 100
6630 IF IN=IN+1 ELSE 100
6640 IF IN=IN+1 ELSE 100
6650 IF IN=IN+1 ELSE 100
6660 IF IN=IN+1 ELSE 100
6670 IF IN=IN+1 ELSE 100
6680 IF IN=IN+1 ELSE 100
6690 IF IN=IN+1 ELSE 100
6700 IF IN=IN+1 ELSE 100
67
```

Listings



Michael Peter Hammer, M.D.

[illegible][illegible]

Year	Population	Population	Population	Population
1990	100	100	100	100
1991	100	100	100	100
1992	100	100	100	100
1993	100	100	100	100
1994	100	100	100	100
1995	100	100	100	100
1996	100	100	100	100
1997	100	100	100	100
1998	100	100	100	100
1999	100	100	100	100
2000	100	100	100	100
2001	100	100	100	100
2002	100	100	100	100
2003	100	100	100	100
2004	100	100	100	100
2005	100	100	100	100
2006	100	100	100	100
2007	100	100	100	100
2008	100	100	100	100
2009	100	100	100	100
2010	100	100	100	100
2011	100	100	100	100
2012	100	100	100	100
2013	100	100	100	100
2014	100	100	100	100
2015	100	100	100	100
2016	100	100	100	100
2017	100	100	100	100
2018	100	100	100	100
2019	100	100	100	100
2020	100	100	100	100
2021	100	100	100	100
2022	100	100	100	100
2023	100	100	100	100
2024	100	100	100	100
2025	100	100	100	100
2026	100	100	100	100
2027	100	100	100	100
2028	100	100	100	100
2029	100	100	100	100
2030	100	100	100	100
2031	100	100	100	100
2032	100	100	100	100
2033	100	100	100	100
2034	100	100	100	100
2035	100	100	100	100
2036	100	100	100	100
2037	100	100	100	100
2038	100	100	100	100
2039	100	100	100	100
2040	100	100	100	100
2041	100	100	100	100
2042	100	100	100	100
2043	100	100	100	100
2044	100	100	100	100
2045	100	100	100	100
2046	100	100	100	100
2047	100	100	100	100
2048	100	100	100	100
2049	100	100	100	100
2050	100	100	100	100
2051	100	100	100	100
2052	100	100	100	100
2053	100	100	100	100
2054	100	100	100	100
2055	100	100	100	100
2056	100	100	100	100
2057	100	100	100	100
2058	100	100	100	100
2059	100	100	100	100
2060	100	100	100	100
2061	100	100	100	100
2062	100	100	100	100
2063	100	100	100	100
2064	100	100	100	100
2065	100	100	100	100
2066</				

[illegible][illegible]

LISTINGS

[illegible]

Binders

Organize and protect your disk with Commodore Disk User disk binders and data disks.

Why not keep your Commodore Disk User program collections alongside your magazines in a single Disk User binder? This binder starts complete with 10 disk drives to organize and protect your program disks. Why not buy a disk binder to house all of your disk data? The disk event supply Commodore Disk User data disks. The Commodore Disk User logo immediately identifies your disks and there's room to title them and document the disk details. Send for more disks and binders now!

Process type: *in vitro*

Continued from Page 14

Order code: **MBF01**

Commanders: Doug Liver, Barker with 10 others, and

Order code: **W00013**

© 2000, for information on ordering, call 1-800-394-3663. © 2000, for information on ordering, call 1-800-394-3663.

30 classes for inclusion in binder. ☐ Yes ☐ No Order code:

■ **Communications:** Daily. User data daily. 45% Online

All orders should be sent to **YOUR COMMODITY, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST**. Please allow 28 days for delivery.



PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Overseas postage add £1.00			
GENE READERS SERVICES 25, WILSON, BRUNSWICK ROAD BIRMINGHAM B15 2JY			TOTAL

Specialists in Mail Order

WHY MEGALAND • 24 Hour Delivery • 12 months warranty on all hardware

• On site maintenance optional • Technical support

• 24 Hour ordering service • Special price offers

• MEGASOFT Club membership for extra special OFFERS

**MEGA
SUPER
OFFER**

**PCI
EXPANSION BOX
£89 + VAT**

BOOKINGS

SPECIAL OFFER

**PCI COMPUTER • HIGH
RES MONITOR
ONLY £309 INC VAT**

**MEGA
SUPER
OFFER**

**TIME
WORKS FOR
D.T.F.
£89 INC VAT**

BOOKINGS

HOW IN STOCK
AMIGA 486 AND 586 PC
REGULAR PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE
BOOKINGS ONLY 1/2 PRICE

AMSTRAD PC 2088

PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00

AMSTRAD PCW

PCW1088	£179.00
PCW1088	£179.00
PCW1088	£179.00

AMSTRAD PPC PORTABLES

PPC1088	£179.00
PPC1088	£179.00
PPC1088	£179.00
PPC1088	£179.00

AMSTRAD PC1512

PC1512 00 01 00	£179.00
PC1512 00 01 00	£179.00
PC1512 00 01 00	£179.00
PC1512 00 01 00	£179.00

AMSTRAD PC1610

PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00
PC1610 00 01 00	£179.00

COMMODORE PRICE REDUCTIONS

FULLY COMPATIBLE SINGLE & DOUBLE
PC EXPANSION BOX

PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00
PC1088	£179.00

COMMODORE DESK TOP PCs

ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00
ALL AT 1/2 PRICE	£179.00

COMMODORE PC10 ATs

PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00

COMMODORE PC 10 3/5 DESK TOP

PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00
PC1088 00 01 00	£179.00

PC1088 00 01 00
PC1088 00 01 00
PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

PC1088 00 01 00

42 - 44 MILLBROOK ROAD EAST, SOUTHAMPTON

OPEN MONDAY TO SATURDAY 9.00am to 5.30pm

AMIGA 500 MEGA PACK £369 INC

INCLUDING MOUSE, WORKBOOK 1.3, UTILITIES, MAILMAIL, BASIC TUTORIAL, PLUS FREE TV INSTALLATION, JOYSTICK AND ANY TEN FREE GAMES FROM OUR SELECTION BELOW

Consistent, Available, Quiet, Spilling Image, 161 Screen, Wider, Glyndaf 55, Helms, Backlight, Dynamic Return of the Jack, Glendale, Donnell, Grindell, Monitory, Southern, Star Screen, Pinnapple, Pinnapple, Pinnapple, Pinnapple

AMIGA HARDWARE

4088 + TV MON	£299 inc
4088 + MONITOR	£299 inc
4088 + MONITOR	£299 inc
4088 + MONITOR	£299 inc
4088 + MONITOR	£299 inc
4088 + MONITOR	£299 inc
4088 + MONITOR	£299 inc
4088 + MONITOR	£299 inc

AMIGA DRIVES

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc

AMIGA ACCESSORIES

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

COMMODORE CBI

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

AMIGA/3000 MONITORS

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

A2000 HARDWARE

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

A2000 ACCESSORIES

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

DRIVES

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

PRINTERS

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

COLOUR PRINTERS

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

LASER PRINTERS

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

MONITORS

4088 + 1 DRIVE	£299 inc
4088 + 2 DRIVE	£299 inc
4088 + 3 DRIVE	£299 inc
4088 + 4 DRIVE	£299 inc
4088 + 5 DRIVE	£299 inc
4088 + 6 DRIVE	£299 inc
4088 + 7 DRIVE	£299 inc
4088 + 8 DRIVE	£299 inc

MEGA SUPER OFFER

with XT 8-Found
25 MB Hard Drive
100MB Colour Monitor
£1399 + VAT

SPECIAL OFFER

Amiga 5000 1.3 WB
100MB(s) Colour Monitor
£879 inc VAT

MEGA SUPER OFFER

Amiga 500 1.3
Monitor 100MB 5
RAM Expansion
£299 inc

SPECIAL OFFERS

EPSON LQ600
24 PIN PRINTER
£399 inc VAT

WESTERN DIGITAL
HARD CARD 32MB
£259 inc VAT

CREDIT CARD MAIL ORDER
SPECIAL HOTLINE FOR EXPORT
TEL 0703 332225
ALL OTHERS SUBJECT TO AVAILABILITY
OFFERS SUBJECT TO CHANGE IN PRICE
WITHOUT NOTICE



ALL PAYERS PLEASE FILL IN THIS FORM

The 40-42 Millbrook Road, Southampton

I wish to order

Or charge my Account/VISA No

Exp date

Signature

Name

Address

Postcode

Tel No

Extending Basic

Create your own-symbol table, and learn all about tables and search routines

By Burghard-Henry Lehmann

In the last article of this series, we developed a routine which allows us to declare labels for GOTOs and GOSUBs in Basic. We did this with a routine that stores each label and its value in the Basic variable area which follows directly after the Basic textfile.

But with this method we are subject to the limitations Commodore Basic imposes on the use of variables: the most obvious being that only the first two characters of a variable name are taken into account—the rest are ignored.

What we want to do is overcome this limitation by creating our own symbol table. In the process we will learn about several concepts in machine-code programming which will be very useful in whatever programming you want to do.

The Symbol Table

The symbol table we want to create is an area in memory where we store each label, and next to it the line number to which the computer has to jump when the label is used with a GOTO or GOSUB instruction.

In machine-code programming, you create many tables. A table is an array of data where each entry is of the same length. Tables are useful for storing data and accessing it quickly and efficiently. For the symbol table we create in our program, I have chosen a memory bank starting from location 50000 (you may of course put the symbol table anywhere you like). Thus 50000 is the base address of our symbol table.

To make each entry the same length, I've chosen a maximum of six recognizable characters for each label. That is, if a label is longer than six characters, the rest will be ignored, and if it's shorter than six characters, the difference will be filled (or "padded") with zeros in the symbol table as you will see in a minute.

Since we also want to store the line number belonging to each label next to the label, each entry needs an

additional two bytes—one byte to store the low byte of the line number, and one byte to store its high byte. This becomes important if the line number is larger than 255.

In all, we use eight bytes for each label stored in the symbol table. Therefore, to point at the beginning of a label, we increment the base address of the table in multiples of eight. And if you call the base address of the table pointer B, then the offsets for the first four entries, for example, on the table are 0, 8, 16 and 24.

Now you should understand why it is important that each entry on a table is the same length. If the computer has to search the table from beginning to end, it can very easily jump from entry to entry simply by adding the length of the entry to pointer address. And if it has to jump to a specific entry straight away, it can do so by adding the offset of that entry to the base address of the table.

Storing a Label

The routine to store each label in the symbol table (lines 2610-3490) is quite simple before the computer starts searching through the Basic program for all the labels, the base address of the symbol table, which is 50000, is set to zero (page 351/352 (line 220-223)). This allows us to use indirect indexed addressing later on.

When a label has been found, zero page 374/375 is pointing at the beginning of the label in the Basic textfile. STORELABEL (lines 3030-3990) stores each letter of the label in the symbol table, using indirect indexed addressing.

If the label is shorter than six characters, the following loop (line 3060-3100) fills the rest with spaces. Next the line number belonging to the label is stored. To do this we increment zero page 351/352, which stores the pointer to the symbol table, by six (line 3140-3190).

Now we come up against a little

difficulty: we have to use indirect-Y addressing for the source and the destination, but Y has to contain a different value. To point at the source, it has to contain two and three, while to point at the destination it has to contain zero and one.

To solve this problem, table the low byte of the line number first and push it onto the stack (lines 3230-3290). Then get the high byte, change the index to one and store it in the symbol table (lines 3290-3299).

Lastly decrement that index by one, pull the low byte of the line number from the stack, and store it in the next free place in the symbol table (lines 3300-3330). Such tricks are necessary to overcome the lack of registers the 6500 has got! (Remember, we can't use the X-register here, because X can only be used to index absolute addresses!)

Finally, increment the symbol table pointer address by another two bytes to make it point at the location where the next label has to be stored.

End Marker

When all labels have been collected and entered into the symbol table, enter a zero in the symbol table, where the beginning of the next label would have been (line 3580-3590).

That is important, because when the computer has to search for a label, it has to know where the end of the symbol table is. Because, if it reaches the end of the symbol table, that means that it hasn't found the label and that the search is finished. If there was no way to determine the end of the symbol table, the computer would go on searching for ever—that is, it would get lost in an endless loop!

To signify the end of a table, it is sometimes useful to put an end marker as we do here. At other times it is more useful to store the end location on the beginning of the last entry in a variable. You have to decide which method is best according to what each you in the specific routine you are writing.

Searching for a Label

When executing the Basic program itself every time the computer needs a label after a GOTO or GOSUB command it has to search through the symbol table to find that label and get the line number next to it which it needs to jump to the right destination.

In order to write a search routine you have to ask yourself some very important questions: when is the search successful, and when is it not? This sounds trivial and obvious, but remember, the computer is a very simple-minded animal and knows nothing about what you have in mind! So you have to define things very accurately. This is half the art of programming a computer.

The search for a label is successful when all the six recognized characters of a label in the symbol table match with the characters of a label in the symbol table. And the search is unsuccessful when the computer has reached the end of the symbol table and hasn't found the label it was searching for.

Once we've got this clearly in our mind, the construction of the search routine itself is not too difficult. The main search loop (lines 1400-1470) compares each character of the label in the symbol table with each character of a particular entry in the symbol table.

It has three tasks

First, number one is taken if the end of the label in the symbol has been reached. This is signalled by a zero (if we make the rule that a label has to be at the end of a Basic line and that nothing else, including remarks, is allowed after it). If this point has been reached it means that the label has been found, even though, if you want to be a perfectionist, you might want to make this routine more accurate and versatile. You can do this by adding up additional tests if the label is not at the end of a line, and if the label in the symbol is shorter than the label in the symbol table.

The second exit of the main loop is taken when any letter in the symbol does not compare with a letter in the symbol table. This does not necessarily mean that the search is unsuccessful at this point. If the label is shorter than six characters, it could mean that it has been found! So lines 1530-1540 test if the next byte in the symbol table contains a zero. If yes, the label has indeed been found. If not, the search has been unsuccessful up to this point.

The third exit is taken after all six recognized characters have been compared and found matching. This means

of course that the label has been found.

Now the line number after the label is gathered in zero page 114/15 (lines 1740-1840), and then the GOTO routine is executed (line 1850).

If the label so far has not been found, lines 1900-1930 increment the pointer to the symbol table (equivalent to zero page 25/26) by eight, so that it points at the beginning of the next label. Then a test is made to see if the

end of the symbol table has been reached (lines 1940-1950).

If it has, the search has been unsuccessful. This means that the computer cannot get a destination line number. Thus line 1950 returns the program flow back to the main routine which results in a syntax error report, signifying that a label in the Basic program has been used which cannot be found.

11	ORG 40104	800	LAF 1
12	END	810	END NORMAL
13		820	J%5 CHANGE
14	CHARACT	830	CMF 0000
15	RECHACT		1 CM FOLDS
16			END COLOR RT
17		900	
18	SEARCH THE	910	
19		920	
20		930	
21		940	END NORMAL END ROUTINE
22		950	
23	TOOK AFTERWARDS BASIC ON	960	NORMAL JMP FOUND
24	BY QUANTUM VECTOR AT	970	
25	80700	980	
26	RETRASON	990	EXECUTE 'COLOR' COMMAND
27	STA 80800	1000	
28	100	1010	GET THE PAROXYSM
29	100	1020	
30	100	1030	COLOR RT
31	100	1040	JMP CHANGE
32	100	1050	JMP 80800
33	100	1060	JMP 80700
34	100	1070	
35	100	1080	CHANGE LINE COLOR
36	100	1090	
37	100	1100	END 840
38	100	1110	
39	100	1120	GET PAPER PARAMETER
40	100	1130	
41	100	1140	JMP CHANGE
42	100	1150	JMP 80800
43	100	1160	JMP 80700
44	100	1170	
45	100	1180	CHANGE PAPER COLOR
46	100	1190	
47	100	1200	END 840
48	100	1210	
49	100	1220	GET BORDER PARAMETER
50	100	1230	
51	100	1240	JMP CHANGE
52	100	1250	JMP 80800
53	100	1260	JMP 80700
54	100	1270	
55	100	1280	CHANGE BORDER COLOR
56	100	1290	
57	100	1300	END 840
58	100	1310	
59	100	1320	JUMP TO NEXT OF ROM
60	100	1330	ROUTINE
61	100	1340	END
62	100	1350	
63	100	1360	EXECUTE 'GOTO' COMMAND
64	100	1370	
65	100	1380	GOTO RT
66	100	1390	JMP CHANGE
67	100	1400	JMP 80800
68	100	1410	JMP 80700
69	100	1420	
70	100	1430	GOTO RT
71	100	1440	JMP CHANGE
72	100	1450	JMP 80800
73	100	1460	JMP 80700
74	100	1470	
75	100	1480	SEARCH FOR LABEL
76	100	1490	
77	100	1500	INITIALIZE START OF SYMBOL
78	100	1510	TABLE
79	100	1520	GOTO RT
80	100	1530	END 840

1740	LDX ZERO	2140	FILE NO	JOB CHARACTER	2540	STORE LABEL
1750	LDX #10000	2150	END OF FILE	END OF FILE	2550	
1760	LDX ZERO	2160			2560	IF #0
1770		2170	END		2570	STORELOC LDX (1000) 0
1780	MAIN SOURCE LOOP	2180			2580	STX (1000) 0
1790		2190			2590	INC LABELNO
1800		2200	MODIFIED 'MM'-EXTENSION		2600	IF
1810	COMPLXOP LDX (1000) 0	2210			2610	OF 40
1820	REQ POSITIVE	2220	JOB 'MM'		2620	END STORELOC
1830	CMF (1000) 0	2230			2630	
1840	REQ NEGATIVE	2240	END OF	LDX #0	2640	
1850	IF	2250	JOB (1000)		2650	REQ POINTNO
1860	CMF #0	2260	JOB (1000)		2660	
1870	END COMPLEX	2270			2670	IF LABEL NUMBERED FROM 0
1880		2280	INITIALIZE SYMBOL TABLE		2680	OR 0
1890	REQ POSITIVE	2290			2690	FILL MAPS OF SPACE WITH
1900		2300			2700	ZERO
1910	SEE IF LABEL NUMBER	2310	LDX #SYMBOLNO		2710	
1920	FROM 0 THEN	2320	STX (1000)		2720	LABELNO LDX #0
1930	NEGATIVE LDX (1000) 0	2330	LDX ADDRESS OF TEL		2730	STX (1000) 0
1940	REQ POSITIVE	2340			2740	IF
1950		2350	GO THROUGH LDC &		2750	OF 40
1960		2360	ADDRESS LABELS		2760	END LABELNO
1970		2370			2770	
1980	IF NOT POINT TO	2380			2780	POINT AT WORD FROM IN
1990	BEGINNING OF	2390	LDX #10000		2790	FROM 100
2000	TABLE	2400	STX 100		2800	
2010		2410	LDX #10000		2810	INITIALIZE LDC
2020	CMF	2420	STX 100		2820	LDX #0
2030	LDX 100	2430	STX 100		2830	END 100
2040	LDX 100	2440	STX 100		2840	END 100
2050	LDX 100	2450	STX 100		2850	END 100
2060	LDX 100	2460	STX 100		2860	END 100
2070	LDX 100	2470	STX 100		2870	END 100
2080	LDX 100	2480	STX 100		2880	END 100
2090	LDX 100	2490	STX 100		2890	END 100
2100	LDX 100	2500	STX 100		2900	END 100
2110	LDX 100	2510	STX 100		2910	END 100
2120	LDX 100	2520	STX 100		2920	END 100
2130	LDX 100	2530	STX 100		2930	END 100
2140	LDX 100	2540	STX 100		2940	END 100
2150	LDX 100	2550	STX 100		2950	END 100
2160	LDX 100	2560	STX 100		2960	END 100
2170	LDX 100	2570	STX 100		2970	END 100
2180	LDX 100	2580	STX 100		2980	END 100
2190	LDX 100	2590	STX 100		2990	END 100
2200	LDX 100	2600	STX 100		3000	END 100
2210	LDX 100	2610	STX 100		3010	END 100
2220	LDX 100	2620	STX 100		3020	END 100
2230	LDX 100	2630	STX 100		3030	END 100
2240	LDX 100	2640	STX 100		3040	END 100
2250	LDX 100	2650	STX 100		3050	END 100
2260	LDX 100	2660	STX 100		3060	END 100
2270	LDX 100	2670	STX 100		3070	END 100
2280	LDX 100	2680	STX 100		3080	END 100
2290	LDX 100	2690	STX 100		3090	END 100
2300	LDX 100	2700	STX 100		3100	END 100
2310	LDX 100	2710	STX 100		3110	END 100
2320	LDX 100	2720	STX 100		3120	END 100
2330	LDX 100	2730	STX 100		3130	END 100
2340	LDX 100	2740	STX 100		3140	END 100
2350	LDX 100	2750	STX 100		3150	END 100
2360	LDX 100	2760	STX 100		3160	END 100
2370	LDX 100	2770	STX 100		3170	END 100
2380	LDX 100	2780	STX 100		3180	END 100
2390	LDX 100	2790	STX 100		3190	END 100
2400	LDX 100	2800	STX 100		3200	END 100
2410	LDX 100	2810	STX 100		3210	END 100
2420	LDX 100	2820	STX 100		3220	END 100
2430	LDX 100	2830	STX 100		3230	END 100
2440	LDX 100	2840	STX 100		3240	END 100
2450	LDX 100	2850	STX 100		3250	END 100
2460	LDX 100	2860	STX 100		3260	END 100
2470	LDX 100	2870	STX 100		3270	END 100
2480	LDX 100	2880	STX 100		3280	END 100
2490	LDX 100	2890	STX 100		3290	END 100
2500	LDX 100	2900	STX 100		3300	END 100
2510	LDX 100	2910	STX 100		3310	END 100
2520	LDX 100	2920	STX 100		3320	END 100
2530	LDX 100	2930	STX 100		3330	END 100
2540	LDX 100	2940	STX 100		3340	END 100
2550	LDX 100	2950	STX 100		3350	END 100
2560	LDX 100	2960	STX 100		3360	END 100
2570	LDX 100	2970	STX 100		3370	END 100
2580	LDX 100	2980	STX 100		3380	END 100
2590	LDX 100	2990	STX 100		3390	END 100
2600	LDX 100	3000	STX 100		3400	END 100
2610	LDX 100	3010	STX 100		3410	END 100
2620	LDX 100	3020	STX 100		3420	END 100
2630	LDX 100	3030	STX 100		3430	END 100
2640	LDX 100	3040	STX 100		3440	END 100
2650	LDX 100	3050	STX 100		3450	END 100
2660	LDX 100	3060	STX 100		3460	END 100
2670	LDX 100	3070	STX 100		3470	END 100
2680	LDX 100	3080	STX 100		3480	END 100
2690	LDX 100	3090	STX 100		3490	END 100
2700	LDX 100	3100	STX 100		3500	END 100
2710	LDX 100	3110	STX 100		3510	END 100
2720	LDX 100	3120	STX 100		3520	END 100
2730	LDX 100	3130	STX 100		3530	END 100
2740	LDX 100	3140	STX 100		3540	END 100
2750	LDX 100	3150	STX 100		3550	END 100
2760	LDX 100	3160	STX 100		3560	END 100
2770	LDX 100	3170	STX 100		3570	END 100
2780	LDX 100	3180	STX 100		3580	END 100
2790	LDX 100	3190	STX 100		3590	END 100
2800	LDX 100	3200	STX 100		3600	END 100
2810	LDX 100	3210	STX 100		3610	END 100
2820	LDX 100	3220	STX 100		3620	END 100
2830	LDX 100	3230	STX 100		3630	END 100
2840	LDX 100	3240	STX 100		3640	END 100
2850	LDX 100	3250	STX 100		3650	END 100
2860	LDX 100	3260	STX 100		3660	END 100
2870	LDX 100	3270	STX 100		3670	END 100
2880	LDX 100	3280	STX 100		3680	END 100
2890	LDX 100	3290	STX 100		3690	END 100
2900	LDX 100	3300	STX 100		3700	END 100
2910	LDX 100	3310	STX 100		3710	END 100
2920	LDX 100	3320	STX 100		3720	END 100
2930	LDX 100	3330	STX 100		3730	END 100
2940	LDX 100	3340	STX 100		3740	END 100
2950	LDX 100	3350	STX 100		3750	END 100
2960	LDX 100	3360	STX 100		3760	END 100
2970	LDX 100	3370	STX 100		3770	END 100
2980	LDX 100	3380	STX 100		3780	END 100
2990	LDX 100	3390	STX 100		3790	END 100
3000	LDX 100	3400	STX 100		3800	END 100
3010	LDX 100	3410	STX 100		3810	END 100
3020	LDX 100	3420	STX 100		3820	END 100
3030	LDX 100	3430	STX 100		3830	END 100
3040	LDX 100	3440	STX 100		3840	END 100
3050	LDX 100	3450	STX 100		3850	END 100
3060	LDX 100	3460	STX 100		3860	END 100
3070	LDX 100	3470	STX 100		3870	END 100
3080	LDX 100	3480	STX 100		3880	END 100
3090	LDX 100	3490	STX 100		3890	END 100
3100	LDX 100	3500	STX 100		3900	END 100
3110	LDX 100	3510	STX 100		3910	END 100
3120	LDX 100	3520	STX 100		3920	END 100
3130	LDX 100	3530	STX 100		3930	END 100
3140	LDX 100	3540	STX 100		3940	END 100
3150	LDX 100	3550	STX 100		3950	END 100
3160	LDX 100	3560	STX 100		3960	END 100
3170	LDX 100	3570	STX 100		3970	END 100
3180	LDX 100	3580	STX 100		3980	END 100
3190	LDX 100	3590	STX 100		3990	END 100
3200	LDX 100	3600	STX 100		4000	END 100
3210	LDX 100	3610	STX 100		4010	END 100
3220	LDX 100	3620	STX 100		4020	END 100
3230	LDX 100	3630	STX 100		4030	END 100
3240	LDX 100	3640	STX 100		4040	END 100
3250	LDX 100	3650	STX 100		4050	END 100
3260	LDX 100	3660	STX 100		4060	END 100
3270	LDX 100	3670	STX 100		4070	END 100
3280	LDX 100	3680	STX 100		4080	END 100
3290	LDX 100	3690	STX 100		4090	END 100
3300	LDX 100	3700	STX 100		4100	END 100
3310	LDX 100	3710	STX 100		4110	END 100
3320	LDX 100	3720	STX 100		4120	END 100
3330	LDX 100	3730	STX 100		4130	END 100
3340	LDX 100	3740	STX 100		4140	END 100
3350	LDX 100	3750	STX 100		4150	END 100
3360	LDX 100	3760	STX 100		4160	END 100
3370	LDX 100	3770	STX 100		4170	END 100
3380	LDX 100	3780	STX 100		4180	END 100
3390	LDX 100	3790	STX 100		4190	END 100
3400	LDX 100	3800	STX 100		4200	END 100
3410	LDX 100	3810	STX 100		4210	END 100
3420	LDX 100	3820	STX 100		4220	END 100
3430	LDX 100	3830	STX 100		4230	END 100
3440	LDX 100	3840	STX 100		4240	END 100
3450	LDX 100	3850	STX 100		4250	END 100
3460	LDX 100	3860	STX 100		4260	END 100
3470	LDX 100	3870	STX 100		4270	END 100
3480	LDX 100	3880	STX 100		4280	END 100
3490	LDX 100	3890	STX 100		4290	END 100
3500	LDX 100	3900	STX 100		4300	END 100
3510	LDX 100	3910	STX 100		4310	END 100
3520	LDX 100	3920	STX 100		4320	END 100
3530	LDX 100	3930	STX 100		4330	END 100
3540	LDX 100	3940	STX 100		4340	END 100
3550	LDX 100	3950	STX 100		4350	

Windows on the C64

Using windows on your C64 couldn't be easier!

By F.E. Randall

This program provides all the facilities you'll need to create a window environment for your Basic programs. These facilities can also be used by means of SYS calls typed in from the keyboard. They allow you to specify the size and shape of the windows you require.

When a window is invoked, all the normal screen editing functions are available, but they only operate on the area of the window you have specified. When that window is 'pushed back', the original screen is restored. Up to four windows may be defined, and each may be 'pushed down' in whatever order you determine.

The definition of a window includes its position on the screen, but when 'pushed down' it may be 'dropped' to another position, and it then becomes the new location for that window in subsequent operations.

The areas used by the system are \$C000 to \$C330 for the machine code, and the screen data is saved under the Basic RAM at \$A000 to \$BFFF. The locations \$F9 to \$FE are used for the parameters of the current window, and

must not be disturbed whilst the window is 'pushed down' otherwise the system may crash. The original values in these locations are saved each time a window is 'pushed down', and restored when the last window is 'pushed back'.

To achieve the usual screen editing facilities whilst only operating on the area of the window, the system includes rewritten parts of the Basic routines CHRIN and CHROUT. When a window is extent, the values at \$F034 and \$0038 are changed to \$C768 and \$C080 respectively. Pressing runstop/restore will reset these vectors if you run into difficulties.

How to Use the System

To use the system, you first have to initialize it by SYS \$B176. This call should also be used if there has been an error message since the parameters can be in an indeterminate state after such an event. More about the error messages later.

After initialization, the windows have to be defined by SYS \$B178, a,

b, c, d, e where:

a = the window serial number from 1 to 4. This number is used to 'pull down' the window later.

b = the number of the row on the screen where the top left hand corner of the window is to appear.

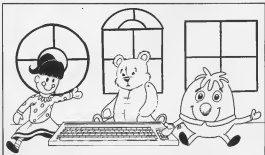
c = the number of the line on which the top left hand corner is to appear.

d = the width of the screen in characters.

e = the number of lines, or the depth of the window.

The window must be a minimum of three characters wide and three lines deep, and the starting row and line plus the width and depth must not exceed 39 and 34 respectively, since these are dictated by the screen's dimensions. In practice, you'll want to leave ample room for any Basic commands you wish to enter from the window.

This brings me to the one exception to the normal screen editing facilities which the system imposes. Normally when a wrap around an input so that each line may be up to 40 characters



long. In the system the input line is interpreted in length to the width of the window you specify. You must make allowance for this when setting the parameters.

The definition of windows will probably come in the initialization problems of your program. If you're going to use them to display prompt messages, you will probably also want to set up the displays in the initialization by "pulling down" the window, PRINTing the text, then "pushing back" the window. "Pulling down" is achieved by SYS 50182, where 'n' is the window number to use in the definition. SYS 50183 will "push back" the window.

Windows can be redefined without restarting the system, but must reinitialize the original contents of the window will be lost. Furthermore, if redefinition takes place too frequently, you may run out of space on which to store the contents of the windows. Better to use the same window for different purposes, and clear it by PRINTing "cl" each time.

Dragging Windows

Having "pulled down" a window, you may drag it around the screen using the following commands:

```
up      - SYS 50181
down    - SYS - 50182
left    - SYS 50184
right   - SYS 50183
```

To drag the window from your Basic program, you would probably want to test for the pressing of a particular function key and then use the appropriate SYS call. For more flexibility, machine code built could "edge" code into the interrupt routine to check whether a function key is pressed and if so, call the relevant subroutine listed above.

Error Messages

The system generates error messages as shown below to help in debugging your program. The conditions are mostly related to the use of incorrect parameters.

Since the system may be left in an underprogram state when the error is detected, after displaying the message, the system waits for a key to be pressed before carrying out a warm start. It's always best to initialize the system again after such an occurrence.

The responses to this are the absence of parameters following the SYS commands, as that is picked up by the Basic interpreter and results in a SYNTAX ERROR message.

The message displayed will take the form "ERR XXX" followed by a letter. The significances of the letters are as follows:

```
a - an attempt made to "pull down"
a window before it has been defined.
b - an attempt made to "pull down"
```

more than four windows at the same window.

c - when defining a window either the start row plus width exceed 16, or the start line plus length exceed 24, or there is no space open to store the window contents.

d - in a window definition, the window number is not in the range one to four.

e - when "pulling down" a window, the window number is not in the range one to four.

f - in a definition, the window width or length is less than three.

Demonstration Program

Included on the disk is a demonstration program introducing the use of windows. Load and run WINDOWS.DEMO. To use the machine code in your own program, type LOAD "WINDOWS.MC", S, I then RUN. The demonstration gives you the choice of "pulling down" windows containing instructions as how to use the system, using a window to view the directory of a disk, defining a window of your own, or "pulling down" one of the windows used in the program. The latter uses window number two, so if you try to "pull down" that window, you will get "ERROR - B".

If you study the listing of the demonstration program, you will see that all is made clear. You may also wish to incorporate the directory listing routine in some of your programs. Happy windowing!

PROGRAM WINDOW DEMO

```

M  10 REM *****
40  10 REM *****
70  10 REM *****
90  10 REM *****
100 10 REM *****
110 10 REM *****
120 10 REM *****
130 10 REM *****
140 10 REM *****
150 10 REM *****
160 10 REM *****
170 10 REM *****
180 10 REM *****
190 10 REM *****
200 10 REM *****
210 10 REM *****
220 10 REM *****
230 10 REM *****
240 10 REM *****
250 10 REM *****
260 10 REM *****
270 10 REM *****
280 10 REM *****
290 10 REM *****
300 10 REM *****
310 10 REM *****
320 10 REM *****
330 10 REM *****
340 10 REM *****
350 10 REM *****
360 10 REM *****
370 10 REM *****
380 10 REM *****
390 10 REM *****
400 10 REM *****
410 10 REM *****
420 10 REM *****
430 10 REM *****
440 10 REM *****
450 10 REM *****
460 10 REM *****
470 10 REM *****
480 10 REM *****
490 10 REM *****
500 10 REM *****
510 10 REM *****
520 10 REM *****
530 10 REM *****
540 10 REM *****
550 10 REM *****
560 10 REM *****
570 10 REM *****
580 10 REM *****
590 10 REM *****
600 10 REM *****
610 10 REM *****
620 10 REM *****
630 10 REM *****
640 10 REM *****
650 10 REM *****
660 10 REM *****
670 10 REM *****
680 10 REM *****
690 10 REM *****
700 10 REM *****
710 10 REM *****
720 10 REM *****
730 10 REM *****
740 10 REM *****
750 10 REM *****
760 10 REM *****
770 10 REM *****
780 10 REM *****
790 10 REM *****
800 10 REM *****
810 10 REM *****
820 10 REM *****
830 10 REM *****
840 10 REM *****
850 10 REM *****
860 10 REM *****
870 10 REM *****
880 10 REM *****
890 10 REM *****
900 10 REM *****
910 10 REM *****
920 10 REM *****
930 10 REM *****
940 10 REM *****
950 10 REM *****
960 10 REM *****
970 10 REM *****
980 10 REM *****
990 10 REM *****
1000 10 REM *****

```

```

4 0 0 0 0 20
10 10 10 10 10
20 10 10 10 10
30 10 10 10 10
40 10 10 10 10
50 10 10 10 10
60 10 10 10 10
70 10 10 10 10
80 10 10 10 10
90 10 10 10 10
100 10 10 10 10
110 10 10 10 10
120 10 10 10 10
130 10 10 10 10
140 10 10 10 10
150 10 10 10 10
160 10 10 10 10
170 10 10 10 10
180 10 10 10 10
190 10 10 10 10
200 10 10 10 10
210 10 10 10 10
220 10 10 10 10
230 10 10 10 10
240 10 10 10 10
250 10 10 10 10
260 10 10 10 10
270 10 10 10 10
280 10 10 10 10
290 10 10 10 10
300 10 10 10 10
310 10 10 10 10
320 10 10 10 10
330 10 10 10 10
340 10 10 10 10
350 10 10 10 10
360 10 10 10 10
370 10 10 10 10
380 10 10 10 10
390 10 10 10 10
400 10 10 10 10
410 10 10 10 10
420 10 10 10 10
430 10 10 10 10
440 10 10 10 10
450 10 10 10 10
460 10 10 10 10
470 10 10 10 10
480 10 10 10 10
490 10 10 10 10
500 10 10 10 10
510 10 10 10 10
520 10 10 10 10
530 10 10 10 10
540 10 10 10 10
550 10 10 10 10
560 10 10 10 10
570 10 10 10 10
580 10 10 10 10
590 10 10 10 10
600 10 10 10 10
610 10 10 10 10
620 10 10 10 10
630 10 10 10 10
640 10 10 10 10
650 10 10 10 10
660 10 10 10 10
670 10 10 10 10
680 10 10 10 10
690 10 10 10 10
700 10 10 10 10
710 10 10 10 10
720 10 10 10 10
730 10 10 10 10
740 10 10 10 10
750 10 10 10 10
760 10 10 10 10
770 10 10 10 10
780 10 10 10 10
790 10 10 10 10
800 10 10 10 10
810 10 10 10 10
820 10 10 10 10
830 10 10 10 10
840 10 10 10 10
850 10 10 10 10
860 10 10 10 10
870 10 10 10 10
880 10 10 10 10
890 10 10 10 10
900 10 10 10 10
910 10 10 10 10
920 10 10 10 10
930 10 10 10 10
940 10 10 10 10
950 10 10 10 10
960 10 10 10 10
970 10 10 10 10
980 10 10 10 10
990 10 10 10 10
1000 10 10 10 10

```

```

1000 10 10 10 10
1010 10 10 10 10
1020 10 10 10 10
1030 10 10 10 10
1040 10 10 10 10
1050 10 10 10 10
1060 10 10 10 10
1070 10 10 10 10
1080 10 10 10 10
1090 10 10 10 10
1100 10 10 10 10
1110 10 10 10 10
1120 10 10 10 10
1130 10 10 10 10
1140 10 10 10 10
1150 10 10 10 10
1160 10 10 10 10
1170 10 10 10 10
1180 10 10 10 10
1190 10 10 10 10
1200 10 10 10 10
1210 10 10 10 10
1220 10 10 10 10
1230 10 10 10 10
1240 10 10 10 10
1250 10 10 10 10
1260 10 10 10 10
1270 10 10 10 10
1280 10 10 10 10
1290 10 10 10 10
1300 10 10 10 10
1310 10 10 10 10
1320 10 10 10 10
1330 10 10 10 10
1340 10 10 10 10
1350 10 10 10 10
1360 10 10 10 10
1370 10 10 10 10
1380 10 10 10 10
1390 10 10 10 10
1400 10 10 10 10
1410 10 10 10 10
1420 10 10 10 10
1430 10 10 10 10
1440 10 10 10 10
1450 10 10 10 10
1460 10 10 10 10
1470 10 10 10 10
1480 10 10 10 10
1490 10 10 10 10
1500 10 10 10 10
1510 10 10 10 10
1520 10 10 10 10
1530 10 10 10 10
1540 10 10 10 10
1550 10 10 10 10
1560 10 10 10 10
1570 10 10 10 10
1580 10 10 10 10
1590 10 10 10 10
1600 10 10 10 10
1610 10 10 10 10
1620 10 10 10 10
1630 10 10 10 10
1640 10 10 10 10
1650 10 10 10 10
1660 10 10 10 10
1670 10 10 10 10
1680 10 10 10 10
1690 10 10 10 10
1700 10 10 10 10
1710 10 10 10 10
1720 10 10 10 10
1730 10 10 10 10
1740 10 10 10 10
1750 10 10 10 10
1760 10 10 10 10
1770 10 10 10 10
1780 10 10 10 10
1790 10 10 10 10
1800 10 10 10 10
1810 10 10 10 10
1820 10 10 10 10
1830 10 10 10 10
1840 10 10 10 10
1850 10 10 10 10
1860 10 10 10 10
1870 10 10 10 10
1880 10 10 10 10
1890 10 10 10 10
1900 10 10 10 10
1910 10 10 10 10
1920 10 10 10 10
1930 10 10 10 10
1940 10 10 10 10
1950 10 10 10 10
1960 10 10 10 10
1970 10 10 10 10
1980 10 10 10 10
1990 10 10 10 10
2000 10 10 10 10

```


[illegible][illegible]

00	7700	0000	0000	0000	0000	0000	0000
01	7710	0000	0000	0000	0000	0000	0000
02	7720	0000	0000	0000	0000	0000	0000
03	7730	0000	0000	0000	0000	0000	0000
04	7740	0000	0000	0000	0000	0000	0000
05	7750	0000	0000	0000	0000	0000	0000
06	7760	0000	0000	0000	0000	0000	0000
07	7770	0000	0000	0000	0000	0000	0000
08	7780	0000	0000	0000	0000	0000	0000
09	7790	0000	0000	0000	0000	0000	0000
10	7800	0000	0000	0000	0000	0000	0000
11	7810	0000	0000	0000	0000	0000	0000
12	7820	0000	0000	0000	0000	0000	0000
13	7830	0000	0000	0000	0000	0000	0000
14	7840	0000	0000	0000	0000	0000	0000
15	7850	0000	0000	0000	0000	0000	0000
16	7860	0000	0000	0000	0000	0000	0000
17	7870	0000	0000	0000	0000	0000	0000
18	7880	0000	0000	0000	0000	0000	0000
19	7890	0000	0000	0000	0000	0000	0000
20	7900	0000	0000	0000	0000	0000	0000
21	7910	0000	0000	0000	0000	0000	0000
22	7920	0000	0000	0000	0000	0000	0000
23	7930	0000	0000	0000	0000	0000	0000
24	7940	0000	0000	0000	0000	0000	0000
25	7950	0000	0000	0000	0000	0000	0000
26	7960	0000	0000	0000	0000	0000	0000
27	7970	0000	0000	0000	0000	0000	0000
28	7980	0000	0000	0000	0000	0000	0000
29	7990	0000	0000	0000	0000	0000	0000
30	8000	0000	0000	0000	0000	0000	0000
31	8010	0000	0000	0000	0000	0000	0000
32	8020	0000	0000	0000	0000	0000	0000
33	8030	0000	0000	0000	0000	0000	0000
34	8040	0000	0000	0000	0000	0000	0000
35	8050	0000	0000	0000	0000	0000	0000
36	8060	0000	0000	0000	0000	0000	0000
37	8070	0000	0000	0000	0000	0000	0000
38	8080	0000	0000	0000	0000	0000	0000
39	8090	0000	0000	0000	0000	0000	0000
40	8100	0000	0000	0000	0000	0000	0000
41	8110	0000	0000	0000	0000	0000	0000
42	8120	0000	0000	0000	0000	0000	0000
43	8130	0000	0000	0000	0000	0000	0000
44	8140	0000	0000	0000	0000	0000	0000
45	8150	0000	0000	0000	0000	0000	0000
46	8160	0000	0000	0000	0000	0000	0000
47	8170	0000	0000	0000	0000	0000	0000
48							



Table 1

[illegible][illegible][illegible]

Star LC10-C Colour Printer

We put the LC10-C under the microscope

By Paul Eves

Shortly after this, I began reversing programs for *Your Commodore*. Yes, you've guessed it, suddenly the need for graphics printing became only too clear. Back went the BII, and I said hello once again to the BII!

Since those early days, I have used many different printers. These machines, like Assemblers, come in all sorts of forms. Some will do one thing, others will do another. It's not often you find a printer that will do **EVERYTHING** you personally want it to do. That is today (and 14 months ago, when I discovered the LC10-C for my money, that has got to be one of the best home-use printers I've ever come across. So what makes this printer stand out above the rest? Well, it's a compact, convenient, colour dot-matrix printer. It's fully compatible with the

in the type! The speed was also better, and to add to all this, it was a lot quieter. To some extent, it was

Ever since I purchased my first computer system, some time and a half years ago, I have always understood the importance of having a printer. Even in those early days as a complete novice, I could see the importance of being able to get hard copies of any important work done be it listings of your latest masterpiece, print-outs of your financials, or simply letters.

The first printer I bought was the BII5001, and at the time it seemed like a good buy. I was able to get print-outs of virtually anything I desired with relative ease. True, the finished result was not spectacular, but it was something - better than my handwriting, I can assure you. However, when I looked around at my friends' listings (lots and lots), I was somewhat miffed to discover that my humble BII was perhaps not quite as good as I had first thought.

Not to be outdone, I tested in the BII and upgraded to the BII. Ah, what noise! This printer had a far superior finish to it. Alright, so it's a lot larger and heavier, but what an improvement

also a lot easier to use. Of course, you can't print out graphics on it, but then again, I thought, I never want to do that anyway. If ever could use the screen in running your program in order to get a print-out of a graphics screen.)



C128 and C64, it supports the Commodore graphics and business character sets, Commodore and IBM versions, and you can also swap to ASCII operating mode.

On top of all that, you have access to no less than ten international character sets. For those of you who like colourful displays and graphics, you have a six colour print option. Operating the printer is simplicity itself. You are presented with a clear, easy-to-read front panel, and from this panel a multitude of operations can be performed.

Apart from the obvious function of switching power on or off, there are the other functions available.

The type style can be selected along with the type pitch. Left and right margins can be set up along with setting the top of paper, and paper can be fed automatically with micro-feed if needed (forward or reverse). Pause printing and the printing of test patterns are both possible.

You can also prevent software from altering your pitch and style settings. One other function, which I think is a bonus to machine code programmers, is the facility for printing a file dump. This is

On top of that, you can have Enhanced, Underlined, Superscript, Subscript, Bold, Double sized and Quadraph and printing. The printer also prints standard eight-bit graphics and Commodore screen-bit graphics.

You can use up to triple-pitch multi-column forms, normal half-fold forms or single sheets. An extra feature is the ability to print half-fold paper, when you are using others. Of course, you don't have to code any word for all that at the end of the review you'll see

But I don't want you to think that I have no criticisms at all, because I have. All that Hi-Tech equipment is somewhat delicate. I had great trouble for some time getting it to respond to colour commands, both from within software and by direct use. I just could not understand what the problem was (the test patterns proved OK).

Eventually I hit on the problem. My ribbon had by this time become bent on Black and Blue. For some reason this interfered with the operation of the rest of the ribbon. On purchasing a new ribbon, the problem disappeared. It would be better if this could be overcome, but I think that this is a small price to pay considering the advantages the machine has.

In the final analysis, I would highly recommend the printer. I know there must be better ones around, the 24 pin type for example, but for all-around versatility, it is hard to beat.

Final Word

The printer that was reviewed here is the Commodore Serial Interfaced Version. There is a Centronics Version that's compatible with Epsons and IBM printers. Therefore this version is suitable with the Amiga and any other computer that supports these printers.

Footnote

Star Electronics, Centre House, 40 Didsbury Rd, Salford M6 2AS. Telephone: (0161) 440 1808.

sample patterns. All these patterns were produced from within simple Basic programs. I think you'll agree that the finished results are quite impressive (especially for a humble home-user printer).

As well as everything else I've said, the supporting manual is well laid out, informative and easy to understand. (It must be if it's understood at all!)

so that you can check that the codes the printer is receiving are correct.

Five type styles are offered for, all of which can be used in tables (one draft and four NLQ). There are then four different print pitches for each.

Spritz Sprite System

*If you are fed up not having sprites on your Plus/4,
read on*

By Mark Everingham

A sprite, according to the Concise Oxford Dictionary is "An evil Fairy or Halpoblin". Human Movie but perhaps the definition should extend to include Cars, Planes, Passes, Invaders from Space and (just for Jeff Minter) Sheep, Cunts and Illusions.

In fact a computer sprite is a picture or any piece of computer graphics which plays an active part in a game. Sadly the Plus/4 does not have any, a fact which has been long lamented by the machine's owners who have to bear the indignity of C64 owners, whose machines are endowed with hardware-controlled sprites.

Spritz sets out to put the balance right by providing your Plus/4 computer with a comprehensive Spritz System, giving not one, not two, but eight independent, full colour sprites which can be controlled very simply through 19 new commands. These commands are added to the standard Commodore Basic language - no need for those engine parts for which the C64 has become infamous!

The Theory Of Sprites

Sprites form the heart of the majority of today's computer games, and are found in abundance on every computer, from the humble C16 to the mighty IBM Armos. A sprite is a graphic element or picture which can be placed on the screen and moved quickly and accurately about with the minimum of trouble.

On the Plus/4, the next screen is

used for sprites because of the speed and ease of use this screen offers. When a sprite is displayed, the computer automatically remembers what was under the sprite before it was printed. To move the sprite you simply change its coordinate position, the computer acknowledges that the sprite has been moved, replaces what was underneath, moves it to the new position and repeats it. *SPRITZ* will detect a collision between a sprite and something else. Sprites can even be printed below or above background graphics.

In addition to these features, *SPRITZ* also has a facility for software priority printing, a feature not found in either the C64 or C128. *SPRITZ* handles the completed program in about 30 milliseconds under the control of an interrupt.

If it all sounds a little technical, remember, all you have to do is tell the computer where to move a sprite, everything else is completely automatic and transparent to the user.

Using The Spritz Commands

SPRITZ adds 19 new commands to basic which makes the control of the sprites simplicity itself, without a FORK in sight. These commands are as exactly the same way as the normal Commodore Basic commands. Every parameter can be expressed either as direct numbers or variables. Standard error messages are given when up, if required, by unaccepting a Basic using the Commodore TRAP command.

The error messages *SPRITZ* gives are explained below.

Syntax Error - You have either used a command or keyword that neither *SPRITZ* or Basic understands, or you have entered too few or too many parameters following a command.

Type Mismatch Error - you have replaced a numeric parameter with a string.

Illegal Quantity Error - One number you have used as a command parameter is outside the valid range for that particular command.

As well as these error messages, the Basic *HELP* command (Key F9), also works with *SPRITZ* commands. If you enter something like -

DEFINE 100, 1, 1, 1, 4, 5, 6, 7, 8, 9
when there should only be the numbers 1-8 the *HELP* command will make all of the last from the final "9" flash to indicate where the error lies.

Similarly, if you typed something like -

SCROLLING RULES ON

The phrase "RULES ON" would be flashing as the valid argument for the *SCROLLING* command is either *ON* or *OFF*. This feature of the *HELP* command with *SPRITZ* makes the location of errors extremely easy.

SPRITZ Set-Up Commands

RAMPONT 1AAAA/AAAA

For the most part, you will want your sprites to be made up of user-defined graphic characters, letters, spaces, etc. You tell *SPRITZ* that you want to use a character set from RAM instead of ROM with the *RAMPONT* command. Its sole parameter is the address where the character set is stored. This can be expressed either in decimal or hexadecimal preceded with a *H* character. *SPRITZ* also uses and manages to return a character set, so when you generate a sprite, the computer will not go wild as it usually does!

Examples
RAMPONT 13000
OR RAMPONT 403D

CEMFONT

However attractive a predefined character set is, it can be difficult to edit a program when all the colours appear as monochrome on the monitor 0-9 as various sections of a computer's body! To get your computer back to the normal Commodore font, just enter *CEMFONT*.

Examples
CEMFONT
OR CEMFONT

PONTCOPY 0/1

Most people find that they only want to redefine part of a character set as user-defined graphics, and still want the usual alphanumeric characters. The answer is to copy the original ROM character set down into RAM and then redefine the characters you want. The *PONTCOPY* command copies a character set into RAM. Either into UPPER case font (*PONTCOPY 0*) or the lower case font (*PONTCOPY 1*). The command must be preceded by a *RAMPONT* command to tell *SPRITZ* where to put the new character set.

Examples
RAMPONT 13000PONTCOPY 0
OR RAMPONT 403DPONTCOPY 1

DEFINE Char. D0, D1, D2, D3, D4, D5, D6, D7

Doesn't it have copied a standard font

into RAM, you will probably want to redefine some of the characters to form graphics with which to build sprites or displays. Usually, this would be done using a series of *DATA* statements and *PEEKS*. These are replaced with one command in *SPRITZ*, *DEFINE*.

The *DEFINE* command takes nine parameters in the range 0-255. The first is the character number, and the 8 that follow are the data bytes which make up the character.

Note, the character number is not ASCII, as in *CHR\$(X)*, but a screen code. The relevant code for each character can be found on Page 162 of your computer manual.

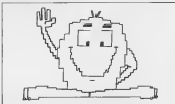
See also the section on the *SPRITZ SCREEN* function.

Example
OR RAMPONT 1300 PONTCOPY 0
20 DEFINE 4, 176, 85, 176, 85, 176, 85, 176, 85
30 PRINT "***"**

HARDHVS and SOFTHVS

A character set can feature no other 128 characters and has the *HVS ON/OFF* function, or 128 characters with no hardware reversing. If you need the extra characters, entering *SOFTHVS* does this, with reversing being handled by software if required. *HARDHVS* selects the lower 128 characters but with *HVS ON/OFF* being handled by hardware.

Examples
HARDHVS PRINT "ABCD
[HVS ON] ABCD"
SOFTHVS PRINT "ABCD
[HVS ON] ABCD"



HIREX and MULTI

The Plus4 computer has several modes, the most important of which are High-resolution and Multi-colour. The *HIREX* and *MULTI* commands switch between the two. *HIREX* mode is usually in operation. *MULTI* mode, which is used to create a number of arcade games, halves the horizontal resolution but allows each pixel to be one of four different colours.

Examples

HIREX PRINT "DRCG SPRITZ"
MULTI PRINT "DRCG SPRITZ"

SPRITZ Sprite Commands

SPRITE SF, CH0, CH1, CH2, CH3, CH4, CH5, CH6, CH7, CH8, CH9

Each sprite under *SPRITZ* control is made up of four characters displayed on the Plus4 low resolution screen. These characters are arranged on a square of 252 characters, and each character can be an individual colour. Reverse or normal characters can be used to make up a sprite, and each colour can be one of the Plus4's full 121. The *SPRITE* command is used to tell *SPRITZ* which characters and colours are to be used to make up a sprite. The parameters it requires are:

SF Sprite number (0-7)
CH0 Top left char code (0-153)
CH1 Top right char code (0-153)
CH2 Bottom left char code (0-153)
CH3 Bottom right char code (0-153)
CH4 Colour for that quarter (1-16)
CH5 Luminance for that quarter (0-7)

As you can see, the characters are specified not as ASCII but as screen codes, as for the **DEFINE** command. The reason for this is that using ASCII gives characters an not unchanged. To specify a reverse character using the characters code you simply add 128 to the original value. Again, the relevant values can be found on page 162 of your manual, also refer the **SPRITE** **SCREEN** section.

Colors are represented using the colour and luminance values that you would use in a **COLOR** command to represent all 121 colours. These are entered in the same order as the character codes for each quarter of the sprite. Each sprite is numbered 0-3 for each of the eight available.

Examples

```
SPRITE 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1248, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1578, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1598, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1648, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1658, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1668, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697, 1698, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1707, 1708, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716, 1717, 1718, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1727, 1728, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1738, 1739, 1740, 1741, 1742, 1743, 1744, 1745, 1746, 1747, 1748, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756, 1757, 1758, 1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777, 1778, 1779, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1787, 1788, 1789, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1798, 1799, 1800, 1801, 1802, 1803, 1804, 1805, 1806, 1807, 1808, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868, 1869, 1870, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879, 1880, 1881, 1882, 1883, 1884, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1892, 1893, 1894, 1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911, 1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 
```


position. When you start an **SPRITE** the sprites are arranged in a row in the top left of the screen.

Example

```
ENABLE * DO MOVE 1 TO X, Y,
ENABLE !
```

SHIFT SP **SHIFT/SCROLL/SUP**
SDOWN and **SHIFT SP** **DOWN**

Usually you will want to move a sprite one character at a time, and the **SHIFT** command can be used to do just this. It will move one sprite one character Left, Right Up or Down. If a sprite goes off an edge of the screen, it reappears at the opposite side. Moving all the sprites to the right makes the sprite reappear on the left. The **SHIFT** command has two possible options. Both take three first parameters: the sprite number, in the range 0-7. Following the sprite number is the direction in which the sprite is to be moved. If the direction is expressed as a number, it should be preceded by a comma. The effect of each of the possible arguments are as follows:

NUMBER	WORD	DIRECTION
0	SHIFT	LEFT
1	SHIFT	RIGHT
2	SUP	UP
3	SDOWN	DOWN

Note that when you specify the direction as English as a single word, the word should be preceded by 'S'. This is to avoid clashes with the BASIC functions LEFTS and RIGHTS.

Example

```
ENABLE * DO SHIFT *
SPRINT/SHIFT/SDOWN LOOP
ENABLES DO - D - RND(1)*4
SHIFT/UP LOOP
```

SPRITZ UTILITY COMMANDS

SCROLLING ON/OFF

One of the problems implementing sprites using software is that whenever a program or user in direct mode causes the screen to scroll up the sprites will also scroll, leaving an image above them. Usually this will not be a problem when programming, but if you wish to print on the bottom line of the screen while sprites are enabled, it can be useful to disable the screen



scrolling effect. To do this just type **SCROLLING OFF** or to reenable it, type **SCROLLING ON**.

When scrolling is disabled, if the cursor is pushed off the bottom line of the screen instead of scrolling the screen up, a just reappears at the top of the screen. The **SCROLLING** command is the equivalent of the ESC "M" and ESC "L" sequences.

Example

```
SCROLLING ON: ENABLE *
CHARI,25,24,""
```

SCREEN SHUTDOWN

Sometimes you may want to scroll the screen up or down but leave the sprites you have enabled intact. The **SCREEN** command will do just this.

SCREEN works by first clearing all the sprites from the screen, scrolling the screen either up or down, and then replotting all the sprites previously enabled. When using many sprites this inevitably causes some screen flicker as it is not really recommended for rapid use. However, it can be very useful when editing programs or experimenting an entirely new effect. **SCREEN SUP** scrolls the screen up and **SCREEN SDOWN** scrolls it down. Each scrolling is one line in each direction. Note that if you have a window specified, the **SCREEN** command will scroll the window only.

Example

```
DO MOVE * TO 24,22
REENABLE *
DO CHARL(RND(4)*40,4,"")
DO SCREEN SHDOWN
DO GOTO 20
```

RESET

If you do something like enter an incorrect address in a **HAMPOINT** command, or leave all your sprites enabled when entering direct mode to

edit a program, it can be hard to tell what's going on when the character set is corrupted. If you get into such a situation entering **RESET** will reset all **SPRITZ** variables to their default values.

The command can also be used at the start of a program to make sure that all the system variables contain the default values. These default settings are shown below -

MODE	FILES
POINT	CENPOINT (Upper case)
REVERSE	SOFTSVS
SPRITES	DISABLED

Example

```
DO RESET/SCREEN: ENABLE *
RESET/SCREEN
```

SPRITE

As **SPRITZ** adds 29 new keywords to the Basic operating system, it can be a little difficult to remember all of them. If you cannot remember a command or keyword, typing **SPRITZ** will display a list of them on the screen. It will automatically configure the list width to the size of the current window.

Example

```
SPRITZ
```

DO SCREEN/SPRITE

SPRITZ User Function

As well as the 19 commands **SPRITZ** adds to Basic, **SPRITZ** also provides you with four new functions. These make the setup and control of sprites easier. These are used just like Basic functions as an argument of another command. For example - **PRINT SCRN("A")** will print the screen on character code value of the ASC(1) "A" character. Equally **CH("SCRN("A")** will assign this value to the variable CH. As you can see these functions are used as the argument for another command, not as a command by themselves.

SCRN ("CHR")

SCRN is used to convert an ASCII character into its equivalent screen code used in the **DEFINE** and **SPRITE** commands. As with all functions, the single parameter is

enclosed within brackets and, as the argument is a string, it should also be enclosed within inverted commas. The string should only be one character long, but if the string you specify is longer, in the same manner as the Basic ASC function, only the leftmost character is considered. You can also use functions from within other languages as long as they yield the correct type of parameter.

Example

```
DEFINE SCRN ("?", 170, 45, 170,
45, 170, 45, 170, 45)
10 SPRITE 0, CHR(64)
20 PRINT S
```

XPOS (SP) and YPOS (SP)

The XPOS and YPOS functions are used to find the position of any sprite.

The two parameters they take is the sprite number, which should be in the range 0-7. The functions should only really be used when the sprite you are finding the position of is enabled. The values returned are in the range of 0-255 for the X and Y-coordinates respectively.

Example

```
ENABLE MOVE TO XPOS (0),
YPOS (0)
10 PRINT "Position of Sprite 0
"XPOS (0), YPOS (0)
```

COLLIDE (SP)

COLLIDE is an important part of the **SPRITZ** system. It tells you whether or not a sprite (SP) has collided with a character which is foregrounded. It returns a zero for no collision or a one for a collision. Collide can be used with the IF command like any normal function. What constitutes a collision is far any of the characters in the 32 square the sprite occupies, to be a non-space character, greater or equal to the **FORGEGROUND** string. Thus the two strings below are respectively no collision and a collision.

Foregrounded value 1
Sprite 0 at (18,40)

Contents

10 - COLLIDE (0)-0
10

10 - COLLIDE (0)-0
10

Example

```
10 CHAR 1, 0, 0, ">" CHAR 2, 0,
0, " "
20 MOVE TO 20, 10 ENABLE 0-0
30 SHIFT 0, 0 IF COLLIDE (0)
THEN 0-1-0
40 GOTO 30
```

SPRITZ Keyword Abbreviation

This brings us end to the discussion of each of the **SPRITZ** commands and keywords, however, to cut down on typing you can use a simple abbreviation for each keyword. The full and abbreviated forms of each **SPRITZ** keyword are shown below.

FULL KEYWORD	ABBREVIATION
COMFORT	C (32bit-0)
COLLIDE	C (32bit-1)
DEFINE	D (32bit-4)
DISABLE	D (32bit-5)
ENABLE	E (32bit-6)
FOREGROUND	F (32bit-8)
FORGEGROUND	F (32bit-9)
MARKERS	M (32bit-4)
MOUSE	M (32bit-5)
MOVE	M (32bit-6)
MULTI	M (32bit-7)
OVER	O (32bit-8)
RAMPOINT	R (32bit-9)
RESET	R (32bit-0)
SCREEN	S (32bit-1)
SCRN	SCRN
SCROLLING	SCR (32bit-0)
SDOWN	S (32bit-1)
SELECT	S (32bit-2)
SHIFT	S (32bit-3)
SLEEP	S (32bit-4)
SOFTS	S (32bit-5)
SPRITE	S (32bit-6)
SPRITZ	SPRITZ
BRIGHT	S (32bit-7)
SUP	SUP
UNDER	U (32bit-8)
XPOS	X (32bit-9)
YPOS	Y (32bit-0)

As you can see, using abbreviations can save you up to seven characters of typing per command, which for those lines represented on the keyboard, can be quite considerable.

SPRITZ Sprite - Editor Program

As well as the **SPRITZ** Basic commands, I have provided a full-featured

Sprite Editor which can be used to create colours, fonts. Sprites for **SPRITZ** with the minimum of effort. As the program is fully cursor controlled, it is not necessary to use any numbers at all to define a sprite.

The Screen Display

When you run the Sprite Editor program you will see that the screen is divided into seven equal or windows. These windows are used as follows.

The Title Window at the head of the screen simply displays the Sprite-Editor heading.

The Edit Window on the left of the screen is where all the action happens. Last displayed a full-colour enlarged version of the sprite currently being edited. In the Edit Window you can move the cursor and change the status of pixels and colours.

When you first run the Sprite-Editor it will contain a display of sprite 0 and a cross wire cursor in the top left of the window.

Next to the edit window are two windows which contain a normal size picture of the current sprite. The windows look identical but the top one shows the current sprite in colour and the lower one in monochrome.

On the centre-right of the screen is the Dialogue Window which for most of the time displays a spreadsheet of the control keys the Sprite Editor program uses. Whenever you are using a function such as saving a set of sprites or changing colours, all the computer output and human input is displayed in this window.

In the bottom-right corner of the screen is a window which displays the eight sprites which are currently in memory, accompanied by their eight-bit number. These sprites are displayed in colour as in the top display window.

The final window, the Status Window is located beneath the two display windows. On the left is displayed the number of the sprite which is being edited, and on the right is the current editing mode.

When first run, these will be 0 and "A".



The Control Keys

The most important keys the Sprite Editor uses are the cursor control keys. If you press these you can set the cross wire cursor in the Edit window, move around the window.

If you move the cursor over a set pixel then it changes to a white cross on a coloured square; if it is on a next (what) pixel then it appears coloured on a white background. In this way the cursor is always visible and the name of the pixel it covers is still displayed. The Sprite-Editor will prevent you from trying to move outside the Edit Window.

Individual pixels of a sprite can be changed by using the 'A' and 'C' keys to Set and Reset the pixel under the cursor. Any changes you make to a sprite are displayed in colour in the top display window and the sprite window, and in the monochrome in the lower display window.

As well as changing the state of pixels individually, you can make the cursor trace a trail of either Set (coloured) pixels or Reset (what) pixels as you move it around the edit window. To make it leave a Set trail press F1, or to make it leave a Reset trail press F2. These are the equivalents of drawing drawings. If you only want to change pixels individually using 'A' and 'C', then you can press F1 to do so. The current drawing made is shown in the status window. A 'V' indicates the draw or set mode, a 'W' indicates the Erase or Reset mode, and a 'C' indicates that pixels will be unchanged when you move the cursor over them, the Over Mode.

One of the features the Sprite-Editor program offers which is unique to SPARTZ is the colour function.

As explained before, a sprite can have a different colour for each of its four single character quarters. By

pressing 'C' you can change the colour of the quarter-Sprite in which the cursor is currently located. When you press 'C' the program will show the

PROGRAM SPRITE LOADER	
10	SPR 00000000000000000000
20	SPR 00000000000000000000
30	SPR 00000000000000000000
40	SPR 00000000000000000000
50	SPR 00000000000000000000
60	SPR 00000000000000000000
70	SPR 00000000000000000000
80	SPR 00000000000000000000
90	SPR 00000000000000000000
100	SPR 00000000000000000000
110	SPR 00000000000000000000
120	SPR 00000000000000000000
130	SPR 00000000000000000000
140	SPR 00000000000000000000
150	SPR 00000000000000000000
160	SPR 00000000000000000000
170	SPR 00000000000000000000
180	SPR 00000000000000000000
190	SPR 00000000000000000000
200	SPR 00000000000000000000
210	SPR 00000000000000000000
220	SPR 00000000000000000000
230	SPR 00000000000000000000
240	SPR 00000000000000000000
250	SPR 00000000000000000000
260	SPR 00000000000000000000
270	SPR 00000000000000000000
280	SPR 00000000000000000000
290	SPR 00000000000000000000
300	SPR 00000000000000000000
310	SPR 00000000000000000000
320	SPR 00000000000000000000
330	SPR 00000000000000000000
340	SPR 00000000000000000000
350	SPR 00000000000000000000
360	SPR 00000000000000000000
370	SPR 00000000000000000000
380	SPR 00000000000000000000
390	SPR 00000000000000000000
400	SPR 00000000000000000000
410	SPR 00000000000000000000
420	SPR 00000000000000000000
430	SPR 00000000000000000000
440	SPR 00000000000000000000
450	SPR 00000000000000000000
460	SPR 00000000000000000000
470	SPR 00000000000000000000
480	SPR 00000000000000000000
490	SPR 00000000000000000000
500	SPR 00000000000000000000
510	SPR 00000000000000000000
520	SPR 00000000000000000000
530	SPR 00000000000000000000
540	SPR 00000000000000000000
550	SPR 00000000000000000000
560	SPR 00000000000000000000
570	SPR 00000000000000000000
580	SPR 00000000000000000000
590	SPR 00000000000000000000
600	SPR 00000000000000000000
610	SPR 00000000000000000000
620	SPR 00000000000000000000
630	SPR 00000000000000000000
640	SPR 00000000000000000000
650	SPR 00000000000000000000
660	SPR 00000000000000000000
670	SPR 00000000000000000000
680	SPR 00000000000000000000
690	SPR 00000000000000000000
700	SPR 00000000000000000000
710	SPR 00000000000000000000
720	SPR 00000000000000000000
730	SPR 00000000000000000000
740	SPR 00000000000000000000
750	SPR 00000000000000000000
760	SPR 00000000000000000000
770	SPR 00000000000000000000
780	SPR 00000000000000000000
790	SPR 00000000000000000000
800	SPR 00000000000000000000
810	SPR 00000000000000000000
820	SPR 00000000000000000000
830	SPR 00000000000000000000
840	SPR 00000000000000000000
850	SPR 00000000000000000000
860	SPR 00000000000000000000
870	SPR 00000000000000000000
880	SPR 00000000000000000000
890	SPR 00000000000000000000
900	SPR 00000000000000000000
910	SPR 00000000000000000000
920	SPR 00000000000000000000
930	SPR 00000000000000000000
940	SPR 00000000000000000000
950	SPR 00000000000000000000
960	SPR 00000000000000000000
970	SPR 00000000000000000000
980	SPR 00000000000000000000
990	SPR 00000000000000000000
1000	SPR 00000000000000000000

colour will be displayed in the Edit, Display and Sprite windows. The cursor will then draw in the new colour selected.

If you want to start with a sprite normal to a screen, then pressing the 'CLR' (HOME) key will clear the Edit window leaving you with an empty blank space. Care should be taken using this function as you can not then recover your hardware!

All of SPARTZ's eight sprites can be edited. You can change the sprite to be edited by pressing the 'S' key. The dialogue window will then clear and you can enter the sprite in the range 0-7. The new number is displayed in the Status window and the new sprite is drawn in the Edit and Display windows. The Sprite can then be edited just like any other.

When you are happy with your sprite you can save these data disk on tape by pressing 'S'. The dialogue window will prompt you for the save device and you should press 'D' for the Disk or 'T' for Tape. The name under which you wish your collection of eight sprites to be saved should then be entered. The file-name should be no longer than ten characters as SPARTZ adds a five character extension of its own.

You can reload sets of sprites previously saved from the Sprite-Editor by hitting the 'L' key. The program will prompt for device and file-name in exactly the same way as for saving sprites. The sprites will then be reloaded for further editing.

Finally, when you have finished using the Sprite-Editor program, you can get out of it by pressing 'ESC' which will return you to home and reset the SPARTZ system variables to their default settings.

Changing the Sprite-Editor Font Address

When you first use the Sprite-Editor program, it is set up for a character set at \$F800 (the highest possible place on a Plus4) or 64K C16. If you want to change that address, for example, if you only have a 32K C16, then all you have to do is change the RAMPOINT command in line 96 and the ADDRESS declaration in line 100 of the program. Care should be taken not to set the address lower than the end of the Sprite-Editor program if you have not saved a backup copy.

Sprite



PAGAN STATE LEADER

```

14869 DATA 78 04 78 78 04 18 78
14870
14871 DATA 88 12 88 12 88 02 02
14872
14873 DATA 18 88 88 81 12 08 07
14874
14875 DATA 24 78 84 82 88 02 07 08
14876
14877 DATA 08 88 08 88 88 18 78 88
14878
14879 DATA 28 88 18 88 18 82 88 88
14880
14881 DATA 07 88 18 8 78 78 88 18
14882
14883 DATA 28 07 88 88 88 88 17 17
14884
14885 DATA 48 88 12 17 88 8 88
14886
14887 DATA 42 82 84 87 88 87 88 08
14888
14889 DATA 73 84 87 82 78 88 73 78
14890
14891 DATA 84 87 12 88 28 73 84 07
14892
14893 DATA 93 88 83 82 81 88 87 87
14894
14895 DATA 08 78 78 88 88 88 88 88
14896
14897 DATA 12 87 83 88 81 17 17 88
14898
14899 DATA 08 88 88 88 88 17 08 88
14900
14901 DATA 78 88 71 88 87 88 88 88
14902
14903 DATA 80 88 83 88 13 88 18 82
14904
14905 DATA 88 88 88 88 88 88 88 88
14906
14907 DATA 88 88 88 88 88 88 88 88
14908
14909 DATA 88 88 88 88 88 88 88 88
14910
14911 DATA 88 88 88 88 88 88 88 88
14912
14913 DATA 88 88 88 88 88 88 88 88
14914
14915 DATA 88 88 88 88 88 88 88 88
14916
14917 DATA 88 88 88 88 88 88 88 88
14918
14919 DATA 88 88 88 88 88 88 88 88
14920
14921 DATA 88 88 88 88 88 88 88 88
14922
14923 DATA 88 88 88 88 88 88 88 88
14924
14925 DATA 88 88 88 88 88 88 88 88
14926
14927 DATA 88 88 88 88 88 88 88 88
14928
14929 DATA 88 88 88 88 88 88 88 88
14930
14931 DATA 88 88 88 88 88 88 88 88
14932
14933 DATA 88 88 88 88 88 88 88 88
14934
14935 DATA 88 88 88 88 88 88 88 88
14936
14937 DATA 88 88 88 88 88 88 88 88
14938
14939 DATA 88 88 88 88 88 88 88 88
14940
14941 DATA 88 88 88 88 88 88 88 88
14942
14943 DATA 88 88 88 88 88 88 88 88
14944
14945 DATA 88 88 88 88 88 88 88 88
14946
14947 DATA 88 88 88 88 88 88 88 88
14948
14949 DATA 88 88 88 88 88 88 88 88
14950
14951 DATA 88 88 88 88 88 88 88 88
14952
14953 DATA 88 88 88 88 88 88 88 88
14954
14955 DATA 88 88 88 88 88 88 88 88
14956
14957 DATA 88 88 88 88 88 88 88 88
14958
14959 DATA 88 88 88 88 88 88 88 88
14960
14961 DATA 88 88 88 88 88 88 88 88
14962
14963 DATA 88 88 88 88 88 88 88 88
14964
14965 DATA 88 88 88 88 88 88 88 88
14966
14967 DATA 88 88 88 88 88 88 88 88
14968
14969 DATA 88 88 88 88 88 88 88 88
14970
14971 DATA 88 88 88 88 88 88 88 88
14972
14973 DATA 88 88 88 88 88 88 88 88
14974
14975 DATA 88 88 88 88 88 88 88 88
14976
14977 DATA 88 88 88 88 88 88 88 88
14978
14979 DATA 88 88 88 88 88 88 88 88
14980
14981 DATA 88 88 88 88 88 88 88 88
14982
14983 DATA 88 88 88 88 88 88 88 88
14984
14985 DATA 88 88 88 88 88 88 88 88
14986
14987 DATA 88 88 88 88 88 88 88 88
14988
14989 DATA 88 88 88 88 88 88 88 88
14990
14991 DATA 88 88 88 88 88 88 88 88
14992
14993 DATA 88 88 88 88 88 88 88 88
14994
14995 DATA 88 88 88 88 88 88 88 88
14996
14997 DATA 88 88 88 88 88 88 88 88
14998
14999 DATA 88 88 88 88 88 88 88 88
15000

```

```

14999 DATA 08 88 78 88 88 18 88 88
15000
15001 DATA 17 81 88 88 81 28 87 84
15002
15003 DATA 88 88 88 18 88 88 18 88
15004
15005 DATA 01 18 88 84 85 88 88 88
15006
15007 DATA 08 88 02 81 88 88 73 18
15008
15009 DATA 82 81 88 88 88 88 81 81
15010
15011 DATA 08 88 88 18 88 88 8 88
15012
15013 DATA 73 18 88 88 88 88 88 88
15014
15015 DATA 88 88 87 88 88 18 8 88
15016
15017 DATA 88 81 18 18 88 88 28 88
15018
15019 DATA 88 88 88 87 88 88 83 88
15020
15021 DATA 88 88 88 88 88 17 88 42 17
15022
15023 DATA 78 18 88 88 88 88 88 18
15024
15025 DATA 18 88 88 88 88 88 88 88
15026
15027 DATA 18 88 88 88 88 88 88 88
15028
15029 DATA 88 87 88 88 18 18 88 88
15030
15031 DATA 88 88 88 88 88 88 88 88
15032
15033 DATA 28 88 18 87 88 88 88 88
15034
15035 DATA 88 88 88 88 88 88 88 88
15036
15037 DATA 88 88 88 88 88 88 88 88
15038
15039 DATA 88 88 88 88 88 88 88 88
15040
15041 DATA 88 88 88 88 88 88 88 88
15042
15043 DATA 88 88 88 88 88 88 88 88
15044
15045 DATA 88 88 88 88 88 88 88 88
15046
15047 DATA 88 88 88 88 88 88 88 88
15048
15049 DATA 88 88 88 88 88 88 88 88
15050
15051 DATA 88 88 88 88 88 88 88 88
15052
15053 DATA 88 88 88 88 88 88 88 88
15054
15055 DATA 88 88 88 88 88 88 88 88
15056
15057 DATA 88 88 88 88 88 88 88 88
15058
15059 DATA 88 88 88 88 88 88 88 88
15060
15061 DATA 88 88 88 88 88 88 88 88
15062
15063 DATA 88 88 88 88 88 88 88 88
15064
15065 DATA 88 88 88 88 88 88 88 88
15066
15067 DATA 88 88 88 88 88 88 88 88
15068
15069 DATA 88 88 88 88 88 88 88 88
15070
15071 DATA 88 88 88 88 88 88 88 88
15072
15073 DATA 88 88 88 88 88 88 88 88
15074
15075 DATA 88 88 88 88 88 88 88 88
15076
15077 DATA 88 88 88 88 88 88 88 88
15078
15079 DATA 88 88 88 88 88 88 88 88
15080
15081 DATA 88 88 88 88 88 88 88 88
15082
15083 DATA 88 88 88 88 88 88 88 88
15084
15085 DATA 88 88 88 88 88 88 88 88
15086
15087 DATA 88 88 88 88 88 88 88 88
15088
15089 DATA 88 88 88 88 88 88 88 88
15090
15091 DATA 88 88 88 88 88 88 88 88
15092
15093 DATA 88 88 88 88 88 88 88 88
15094
15095 DATA 88 88 88 88 88 88 88 88
15096
15097 DATA 88 88 88 88 88 88 88 88
15098
15099 DATA 88 88 88 88 88 88 88 88
15100

```

```

15099 DATA 78 88 88 87 88 88 88
15100
15101 DATA 18 88 88 88 88 88 88 18
15102
15103 DATA 88 88 88 88 88 88 88 88
15104
15105 DATA 18 88 12 87 87 88 88 87
15106
15107 DATA 88 87 87 88 88 88 88 88
15108
15109 DATA 08 78 88 88 88 88 88 88
15110
15111 DATA 08 78 88 88 88 88 88 88
15112
15113 DATA 08 78 88 88 88 88 88 88
15114
15115 DATA 08 78 88 88 88 88 88 88
15116
15117 DATA 08 78 88 88 88 88 88 88
15118
15119 DATA 08 78 88 88 88 88 88 88
15120
15121 DATA 08 78 88 88 88 88 88 88
15122
15123 DATA 08 78 88 88 88 88 88 88
15124
15125 DATA 08 78 88 88 88 88 88 88
15126
15127 DATA 08 78 88 88 88 88 88 88
15128
15129 DATA 08 78 88 88 88 88 88 88
15130
15131 DATA 08 78 88 88 88 88 88 88
15132
15133 DATA 08 78 88 88 88 88 88 88
15134
15135 DATA 08 78 88 88 88 88 88 88
15136
15137 DATA 08 78 88 88 88 88 88 88
15138
15139 DATA 08 78 88 88 88 88 88 88
15140
15141 DATA 08 78 88 88 88 88 88 88
15142
15143 DATA 08 78 88 88 88 88 88 88
15144
15145 DATA 08 78 88 88 88 88 88 88
15146
15147 DATA 08 78 88 88 88 88 88 88
15148
15149 DATA 08 78 88 88 88 88 88 88
15150
15151 DATA 08 78 88 88 88 88 88 88
15152
15153 DATA 08 78 88 88 88 88 88 88
15154
15155 DATA 08 78 88 88 88 88 88 88
15156
15157 DATA 08 78 88 88 88 88 88 88
15158
15159 DATA 08 78 88 88 88 88 88 88
15160
15161 DATA 08 78 88 88 88 88 88 88
15162
15163 DATA 08 78 88 88 88 88 88 88
15164
15165 DATA 08 78 88 88 88 88 88 88
15166
15167 DATA 08 78 88 88 88 88 88 88
15168
15169 DATA 08 78 88 88 88 88 88 88
15170
15171 DATA 08 78 88 88 88 88 88 88
15172
15173 DATA 08 78 88 88 88 88 88 88
15174
15175 DATA 08 78 88 88 88 88 88 88
15176
15177 DATA 08 78 88 88 88 88 88 88
15178
15179 DATA 08 78 88 88 88 88 88 88
15180
15181 DATA 08 78 88 88 88 88 88 88
15182
15183 DATA 08 78 88 88 88 88 88 88
15184
15185 DATA 08 78 88 88 88 88 88 88
15186
15187 DATA 08 78 88 88 88 88 88 88
15188
15189 DATA 08 78 88 88 88 88 88 88
15190
15191 DATA 08 78 88 88 88 88 88 88
15192
15193 DATA 08 78 88 88 88 88 88 88
15194
15195 DATA 08 78 88 88 88 88 88 88
15196
15197 DATA 08 78 88 88 88 88 88 88
15198
15199 DATA 08 78 88 88 88 88 88 88
15200

```

LISTINGS

[illegible]

[illegible]

Contributions

So you own a Commodore? So you've written some programs? So why haven't you sent them to us?

Your Commodore is always on the look out for new programs, hints and tips, articles and even regular series. In fact if you have something that you think could be of use to other Commodore owners we want to hear about it.

So if you have got something which you think we may be interested in. How do you go about submitting it to us?

Below you will find a list of guidelines that will help us to deal with any area that you raise in an article. We don't expect everybody to be the next William Shatnershire but if you do follow these simple rules then it will make our job a lot easier.

1) If possible all material sent to the magazine should be typed or printed out on a computer printer.

2) All text should be double spaced (or there should be a blank line between each line of text). You should allow a margin of about 10 characters around the text.

3) On the very first page you should put the following:

Name of the article
Machine that it is for
Any extra required - disk, printer etc.
Your name
Your address
Your telephone number

4) The top of every page should have the following information in it:
Abbreviation of the article title
Your name
The page number

For example, suppose you had to include an article on C64 interrupts. You should put something like the following at the head of the page:

Interrupts/T Smith/1

5) Please make sure that you do not make any additional marks on your text especially underlining.

6) Try and write in clear concise English, it does not have to be a work of literature but it must be comprehensible.

7) Use the bottom of each page you should put the word **MORE** if there are more pages to the article or **ENDS** if it is the last page.

8) If possible, enclose a listing of all programs.

9) Under no circumstances use a staple to hold the pages together. Use a paperclip instead.

10) Programs should be included on either disk or tape. Make sure that you **SAVE** two copies of every program so that we have a better chance of loading them if problems occur.

11) Programs under 10 lines can be included in the text. If your program is longer than this you must include a disk or cassette.

12) If your article needs any artwork then supply clear examples of what is needed. We don't expect you to be an artist but we do need to see what is required.

13) Photographs if necessary must be color black and white prints or colour slides. We can take them ourselves so don't worry about this too much.

14) Submissions of any length are welcome. If you have a five line routine that you think may be of use to someone else we welcome it just as much as a full blown ten part series.

15) Payment is not quite a lot and depends on quite a number of factors, such as simplicity of program, presentation of program, number of magazine pages it takes up etc. Payment is generally between £10.00 and £50.00.

16) All payments are made in the month that the magazine containing your article has appeared in print.

17) If we do need your submission suitable for inclusion in the magazine we will write to you giving the terms of publication, the time of payment and an agreement form. (Receipt return of this form will allow us to use your program as soon as possible).

18) If you want the program returning to you, should we find it unsuitable for publication then you should enclose a stamped self addressed envelope.

19) The last and most important point to make is 'get writing' we are waiting for your articles.

THE REAL GHOST

If you came across the original (and presumably, in view of this title, unusual Ghostbusters) game years ago at a computer show I was helping out with, those wonderful people on the Acornware stand played the theme music from the film non-stop for four days. That, coupled with an unbelievable amount of hype and a mediocre product, left my feelings for the original game at best lukewarm.

Now, years later, with the television programme currently proving immensely popular, I was interested to see whether the sequel was going to prove to be a major improvement on its predecessor. Sadly, the game is little more than a glorified shoot-'em-up. You must wander round trying to clear an area of assorted ghosts, ghouls and things that go bump in the night. At the end of the levels there are two or three you have to defeat, a bigger, badder baddy who, upon going to that great haunted house in the sky, lets slip a key allowing you access to the next stage of the game.

You can decide the on-screen effect by shooting them or sucking them with your proton beam. The latter method of

has the advantage that the ghosts can then be used in your backpack for a point bonus when you complete the level. The disadvantage, though, is that this weapon only has a limited amount of energy, whereas your gun has a limitless supply of bullets.

The energy for your proton beam can be topped up at one of a number of bases that can be selected on your travels. You can also maintain your life power, provide yourself with a protective shield or extend the strength of your handy night vision here. The life-restoration is around your head, destroying anything in touch, also offering a form of instant protection.



STBUSTERS

Picking up bones is simply a matter of flicking and then walking over them. Some are hidden under oil drums, some carried by ghosts. The most useful ones though are behind barriers of pits. Things like graphics aren't clear enough to determine what exactly'll run and fall out of the ground. Contact with these spikes is fatal, so you have to aim your run into the precision ring precisely.

The graphics and sounds are very chunky, and even though there is a large variety of ghosts, my overall impression was that the whole thing looked very dated. Handling was reasonably smooth, although there was a fair

amount of screen flickering. The Ghostbuster theme tune has been turned up to include a few new sound effects, but I must own up to not loving the original.

Like the original, *The Real Ghostbuster* is lacklustre. There's nothing here to make you sit up and take notice. It might be worth a second look if you are a fan of the series, but overall, I can't imagine myself coming back to have another go.

File: *The Real Ghostbuster* **Supplier:** Activision **File:** 1989 **Size:** 1488 **Cost:**

TIME THIEF

Here is something of a rarity - a reviewed adventure! At a time when even the mighty Infocom have reluctantly been forced to include graphics in their games, it's nice to find someone who still believes that a picture is not necessarily worth a thousand words, and that the powers of imagination will create for something.

The wizard Rodriguez has created a system of time where travel. Rather than winding away with never old, you can trade some of that time for a decent holiday now. Unfortunately, the vigour of the system means that you can't use up your own years, but somebody else. But if while away their time, you'll be in one up many an available

effort can be created and everybody ends up with what they want.

That at least, is the theory. But Rodriguez refused to become his creation, and made himself more, anyone who wanted a share of the action. Now he has a problem as somebody has managed to penetrate the system and steal some time from the customers so that they are becoming old, withered, even dead.

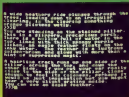
In an attempt to sort out the problem and save himself from financial ruin, the wizard sent his assistant Jacinta to investigate, but she has now disappeared and so it is up to you - but first, to find you can sort things out.

The game plays in two sections, and there is plenty for you to get your teeth into. Location descriptions are very long and atmospheric, and you'll have to read carefully, but I would be reluctant to place the appropriate value. There is a nice touch of whimsical humour running through the story too. The problems and puzzles are again with the storyline, show a considerable amount of imagination and creativity in what has all too often been reduced to a lackluster art form. The game is perfectly adequate, and I found no problems with the vocabulary either.

Time Thief is on disk only and at only five pounds represents excellent value for money. The game is not available through mail order, address below.

Touchline

Title: *Time Thief* **Supplier:** Big Fish Software, 15 Old London Road, Clingwall, Rotherham S68 6RN **Price:** £5.00 **(disk)**



Amiga Workbench

Since its appearance on the market some four years ago the Amiga has gone from strength to strength. Today it is one of the most powerful and versatile home computers available. And yet, recently it has actually gone down in price. So, the band of Amiga owners is growing all the time.

I always maintain that a computer is as good as what you make of it. This is especially true for computers. Nobody has any queries about what to do with a television set or a video recorder or a VCR. Once you've bought it, you just get it out of its box, plug it in, press the ON button, and Bob's Your Uncle.

Computers are very different from that. When you start a computer it does very little at first. All you've got is a closed box of electronic bits and pieces.

This is true even for a machine as powerful as the Amiga. Its real magic is hidden underneath the surface, only unleashed by good software and by your own enthusiasm to get to know the machine and make use of its many capabilities.

Now, when you switch on the Amiga, the first thing you are greeted by is Workbench. And that's a whole lot more than what you get from most other computers at start-up, especially the 3-bit machines we all grew up with like the Commodore 64, the Sinclair Spectrum and the Acorn BBC machines. I don't know if some of you will remember the ancient days of computers when all you were greeted with at start-up was a flashing cursor at some level.

Amazing how far things have come since!

But, Workbench too is merely an environment in which to make things happen on the Amiga. It's not the main event itself.

Workbench is basically an application program that allows you to run other programs and do some disk operations, like disk copy, delete, for making and so on. All the facilities of Workbench are made possible by an important part of the Amiga operating system, called *Intuition*.

Whenever you create a window, pull down a menu, click the padlet on the left hand side of a window which makes the window visible, it's *Intuition*.

By Bernhard-Henry Lehmann

non, not Workbench, that makes it all happen.

Amigaos is a programmer's delight because it puts all the sophisticated facilities of the Amiga right at his fingertips. The programmer does not have to draw lines to form the borders of a window, nor does he have to construct pull-down menus or complicated gadgets. Amigaos does it all for him, and informs him whenever something really important has happened.

When you first try to find out more about the Amiga, and especially how to program it, the task seems quite daunting. This isn't helped by the literature which is supposed to help you like the standard Amiga manuals published by Addison-Wesley. You wonder what kind of computer giant it takes to understand the workings of the Amiga, let alone, to be able to program it.

But once you've overcome the initial barrier and got some insight into the Amiga, you'll discover to your surprise that the Amiga is in fact one of the easiest machines to program!

Yes, that's true! Never mind if you program in Basic or in C or even in machine code: the Amiga puts all its capabilities more easily at the disposal of the programmer than any other machine! Before you know it, your program will have windows and pull-down menus, gadgets and inspectors and will run comfortably in the multitasking environment of the Amiga together with other programs.

To help you on your way to discovering the workings of the Amiga and how to extend its power for your own purposes and desires, *Your Commodore* is starting this new monthly column.

My main aim is that column is to discuss all aspects of Amiga computing with special emphasis on programming the Amiga.

For this I'd like to hear from you! What are your main interests in Amiga computing? What persuaded you to buy an Amiga in the first place? What do you find most difficult and confusing about the Amiga?

In short, send me your opinions, your queries, your suggestions.

But let's start the ball rolling by having a closer look at Workbench 1.2 which most of you will have by now.

If you compare Workbench 1.2 with its predecessor Workbench 1.1 there seems very little difference. The main advantage with Workbench 1.2 is that it runs a bit quicker because of new facilities like fast text and fast flag.

You are also able to rename the Ram disk, something you couldn't do on Workbench 1.2 because there was a bug.

The main advantage with Workbench 1.2 does not lie with Workbench at all, but with CLI. The new shell window makes the real difference to Workbench 1.2.

As far as commands are concerned Shell is very much like CLI, except for two new commands. Its main advantage lies in the vastly improved editing facilities, which has to be a big improvement because the old CLI didn't have any editing facilities at all except backspace.

Shell also more like a screen editor than a line editor. Unlike the CLI it's got a history. That is: You can see the current up and down keys to recall commands you have typed in before. This makes it possible to edit a wrong command you've given instead of having to type it all in from scratch again.

You can also use cursor left and right to correct anything you've typed in wrongly or add something you've forgotten.

Then the Shell has got two new commands which are extremely useful: *Alias* and *Runset*.

This allows you to design your own commands. For example, I have made up for my purposes some aliases to give me the directory of the disk in drive 0 while "d0" gives me the directory of the disk in drive 0 while "d1" gives me the directory of the disk in drive 1. Furthermore, "d2" gives me the directory of the disk in drive 1 with system: while "d3" gives me the same for drive 1.

With the *Alias* command you can use square brackets like *argv* or a filename or variable like *argv* or then be given together with the *Alias*. For example:

```
alias d0 dir d00[
```

gives you, if used by shell, a prompt

of the directory in drive 0 (which is of course the default drive), if you enter "d0 MyDirectory" you get a prompt of "MyDirectory" on the disk in drive 0.

The other new command the Shell has is *Runset*.

This makes a command behave twice as if you had typed a CLI command into the RAM disk so that it executed quicker and without the Workbench shell in the drive.

Runset also works for programs. But there is the restriction that resident only works for CLI commands and programs that are resident and re-entrantable, meaning programs that can be used by several processes at the same time in the Amiga multitasking environment. This is not as restrictive as it seems, since most programs can be made resident quite happily with the "resident pure" option, even if at first it seems to make that program resident. All you have to do is experiment a bit with the program to make sure that the machine won't pile up.

Finally, the Shell has a startup script, called "Shell-startup". You can find it in the subdirectory of the Workbench disk.

In the Shell startup script you can write all your favourite aliases so that they will be automatically installed every time you start a new shell process.

This seems to me to be the main advantage with Workbench 1.1. Workbench 1.2 has a much larger startup-script than Workbench 1.2 and even has a sub-script called *Startup*. All this enables you to set up your own Amiga work environment as it suits you.

To finish off, let me know about any good scripts you've written lately.

Info

Send your queries, hints and general Amiga comments to

Amiga Workbench
Your Commodore
Argus House
Boundary Way
Riverside Hampstead
Hants
HP21 7ST

Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

If it were a clock in the morning. You sit at the computer keyboard having just finished a marathon typing session. Striving, one of the superb programs from Your Commodore. Your fingers reach for the keyboard and press the letters M, U and N. You press RETURN at last and nothing happens.

Everyone has probably lived this problem. When it does happen it's a matter of spending hours searching through the program for one typing mistake. No matter how long you look as how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The Your Commodore Software Service makes available all of the programs from each issue on both cassette and disk at a price of £5.00 for disk and £4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available on the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFOMET LTD, 5 River Park Estate, Bockenhurst, Herts
HP4 1HL
Tel: (04627) 76661

Please contact this address for prices and availability

The Disk

Programs on the disk will also be supplied as totally working versions, ie when possible we will not use lines. Loading then making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 machines. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strip containing the article type, C64 Programs etc. So that you can see which programs are available on which format, you will also find a couple of symbols after the strip. The symbols have the following meaning:



This symbol means that the program is available on cassette



These programs are available on disk

Please Note

Since the programs supplied on cassette are used working versions of the program, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

JANUARY 1987

PREFAB SPRITES - A powerful sprite editor for the C64

DAZPRO - A simple but helpful text processor for the C64. Available on disk and cassette but will only store files on tape

LDC COMPRESSOR - Save on memory when using LDC's in your programs. For C64 only

WILLIAM TELL - Our popular article game for the C64
+4 **AUTOTRYN** - Improve tape loading on your Plus/4 cassette. Disk available on disk

MINIRACE - A database for C128 events

ORDER CODE

DISK TDFJAN87 £5.00

TAPE TDFJAN87 £4.00

FEBRUARY 1987

TAPE MENU - Add a menu system to your program (cassette of £4)

SONIC EFFECTS - A superb sound editor for the C64

F DUMP - Dump your C64 test patterns to printer with ease

DATA LOADER - A simple way to enter those mass of C64 DATA files

SPRITE LIBRARY - A collection of banks to your gaming library (C64)

PLAY THE GAME - A superb first machine program for the Plus/4 (Available on disk only)

ORDER CODE

DISK TDFEB87 £5.00

TAPE TDFEB87 £4.00

MARCH 1989

PERSONAL FILE - A cross between a wordprocessor and a database that allows you to set up "cards" that can be quickly altered (C64 disk only)

LETTER WRITER - An 80 column text editor for writing those personal letters (C64)

BASIC WORKSHOP - A single key entry system, just like a Spaciv (C64)

HEAD FOR HOME - Our version of a popular board game for C16 and Plus/4 owners - available on disk only

SPRITE LIBRARY - Geometric shapes from this month's installment (C64)

ELECTRONIC NOTEBOOK - A personal diary on disk (C64 disk only)

WILLIAM TELL - Our very popular arcade adventure for the C64

ORDER CODE

DISK YDMAR89 64.00

TAPE YCMAR89 64.00

APRIL 1989

BASIC - Our year C64 new sound, graphics and input commands as well as a machine code assembler with this Basic extension

AUTOSCROLL - Professional text screen scrolling with the C64 editor

BALANCE SHEET - Keep your bank manager happy by keeping better track of your money with this C64 program (disk only)

ICON 88 - Add icons control to your own programs with this C64 utility

88 COLUMNS - A suite of Hi-res graphic commands for C128 88 column users

ZEUS - A disk file copier for Plus/4 users. Available on disk only

MULTI-PRECISION ARITHMETIC - A suite of powerful mathematical routines expanding your C64's number crunching abilities

STORAGE SPACE - Use the RAM behind the C64's Kernal and Basic ROMs for data storage (available on disk only)

ORDER CODE

DISK YDAPR89 64.00

TAPE YCAPR89 64.00

MAY 1989

ANTI-FREEZE - Protect your C64 programs from surridge-based freezes

MULTI-COLOURED LIST - Brighten up your C64 Basic listings by adding no-screen colour

PRINTFX - A suite of powerful screen printing commands to add to your Plus/4's Basic (disk only)

BANKER 128 - Money Management for C128 disk users

TURBO TOUCH - A superb typing trainer for C64 disk users

ORDER CODE

DISK YDMAY89 64.00

TAPE YCMAY89 64.00

Customers or debts are available from March 1985. Please ring the editorial office (01-637 0624) for details of these

ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
JULY 89		TAPE (£4.00)	YC JUL79	
JULY 89		DISK (£6.00)	YD JUL79	
OVERSEAS POST £1				
			TOTAL	

NAME

ADDRESS

POSTCODE

I enclose a cheque/postal order for £..... made payable to ABOLUS SPECIALIST

PUBLICATIONS LTD.

All orders should be sent to: YOUR COMMODORE

READERS SERVICES, ABOLUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP1 1ST

Please allow 30 days for delivery

B

A

E

A

H

Readers Problems

Though the Commodore 64 is one of the world's most popular microcomputers, it can be very difficult to find specific information about your particular machine.

At the Four Commodore office we receive literally hundreds of letters from you our readers, on a wide range of subjects ranging from the simple "Can you give me the telephone number for..." to the more complex "I'm trying to write a program that uses a split screen. How do I do it?"

Unfortunately, the volume of mail received has become so great that it is impossible to answer every letter and still manage to publish a magazine each month.

For this reason, we have felt it necessary to produce a number of guidelines for getting information from us.

1) We cannot guarantee to answer every letter sent to the magazine. Should it become apparent that a number of readers are suffering from the same problem, then we will reply to the letter via the Letters page.

2) A new helpline has been set up. This will be open for your questions on

Tuesday and Thursday afternoons between 2:00pm and 4:00pm. We will not be able to deal with our telephone queries at any other time. If our technical advice is not available when you ring, then a message will be taken.

3) If you are having problems with one of our listings, can you please let us know in writing. This will enable us to see if a number of people are having the same problem. When a common problem becomes apparent with a program, then a correction sheet will be issued. Enclosed a self-addressed, stamped envelope and we will send you a copy of the correction sheet as soon as it is available.

We are sorry that it has become necessary to outquote these rules. However, we are sure that you will agree with us that the more time that we can spend making *Your Commodore* the most informative magazine around the better.

For program queries write to:

Program Corrections

Four Commodore

Apex House

Boarders Way

Hemel Hempstead

HP2 7ST

Tel (0440-6631)

Bag Funder

We'd like to remind our readers that we run a Bag Funder service.

If you have typed at one of our programs and dropped something, you still want to get it to run, then send on the following: Two copies of your program on tape or disk.

A description of your problem. If possible a listing of your work (you may omit lines).

A stamped self-addressed envelope for return of the program to you.

Should any of the above be missing, then we will not be able to deal with your query.

We will try to point out where you have made a error and place a corrected copy of the program back on to your tape or disk before we return it to you.

Do not send a program to us as soon as it stops working; please check it several times first.

We do get a large number of queries and so it may take a while for us to deal with yours personally.

Please let us know what you think of our service by writing to us at:

Four Commodore

The Nibbles



By Alan Brinkley

RENEGADE

The leading American data back-up/utility package is now available in the U.K. The world's most powerful data backup package? We are sure that Renegade is the most powerful! For reasons to suit package of its kind. Renegade automatically detects the protection method used - substitutes for tape backup, & copies the data & all the other records.

Some of the features:

- a. Single or dual high speed tape copy
- b. IBM 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000
- c. 9000 module auto 9000 software
- d. 9000 file system - stored under backup
- e. Data backup file unique under backup
- f. 9000 backup file unique under backup
- g. Backup file unique under backup
- h. Backup file unique under backup
- i. Backup file unique under backup
- j. Backup file unique under backup
- k. Backup file unique under backup
- l. Backup file unique under backup
- m. Backup file unique under backup
- n. Backup file unique under backup
- o. Backup file unique under backup
- p. Backup file unique under backup
- q. Backup file unique under backup
- r. Backup file unique under backup
- s. Backup file unique under backup
- t. Backup file unique under backup
- u. Backup file unique under backup
- v. Backup file unique under backup
- w. Backup file unique under backup
- x. Backup file unique under backup
- y. Backup file unique under backup
- z. Backup file unique under backup

The specification & support are impressive. The speed available is better than other backup data in under 10 seconds. Renegade includes over 100 programs, more than any other backup product on the market. It substitutes parameters for changes using one month's delay. Renegade is updated three times a year to ensure that Renegade will always be a step ahead of anything else on the market.

Backup can stop you from having any control of your software. Renegade gives you the most advanced backup data on the market for any reasonably size price. It offers such a complete back door into your information system. Renegade is produced by a company who know that the customer is not their business. Renegade 9000.

JIFFY DOS

Jiffy DOS is an enhancement disk operating system that freely runs your Commodore command line file transfer speed. It was always needed. Programmed into our DOS, the system for format DOS in your computer & the DOS code in your disk. Jiffy DOS provides features that no other product can match. Jiffy DOS is compatible with the majority of computer software & hardware. It includes the most disk updates from Commodore, the latest copy file disk access in both 80 & 115 modes.

Jiffy DOS is available for all these computers: C-40, C-64, C-128, C-128C, C-128D, C-128E, C-128F, C-128G, C-128H, C-128I, C-128J, C-128K, C-128L, C-128M, C-128N, C-128O, C-128P, C-128Q, C-128R, C-128S, C-128T, C-128U, C-128V, C-128W, C-128X, C-128Y, C-128Z, C-129, C-129A, C-129B, C-129C, C-129D, C-129E, C-129F, C-129G, C-129H, C-129I, C-129J, C-129K, C-129L, C-129M, C-129N, C-129O, C-129P, C-129Q, C-129R, C-129S, C-129T, C-129U, C-129V, C-129W, C-129X, C-129Y, C-129Z, C-130, C-130A, C-130B, C-130C, C-130D, C-130E, C-130F, C-130G, C-130H, C-130I, C-130J, C-130K, C-130L, C-130M, C-130N, C-130O, C-130P, C-130Q, C-130R, C-130S, C-130T, C-130U, C-130V, C-130W, C-130X, C-130Y, C-130Z, C-131, C-131A, C-131B, C-131C, C-131D, C-131E, C-131F, C-131G, C-131H, C-131I, C-131J, C-131K, C-131L, C-131M, C-131N, C-131O, C-131P, C-131Q, C-131R, C-131S, C-131T, C-131U, C-131V, C-131W, C-131X, C-131Y, C-131Z, C-132, C-132A, C-132B, C-132C, C-132D, C-132E, C-132F, C-132G, C-132H, C-132I, C-132J, C-132K, C-132L, C-132M, C-132N, C-132O, C-132P, C-132Q, C-132R, C-132S, C-132T, C-132U, C-132V, C-132W, C-132X, C-132Y, C-132Z, C-133, C-133A, C-133B, C-133C, C-133D, C-133E, C-133F, C-133G, C-133H, C-133I, C-133J, C-133K, C-133L, C-133M, C-133N, C-133O, C-133P, C-133Q, C-133R, C-133S, C-133T, C-133U, C-133V, C-133W, C-133X, C-133Y, C-133Z, C-134, C-134A, C-134B, C-134C, C-134D, C-134E, C-134F, C-134G, C-134H, C-134I, C-134J, C-134K, C-134L, C-134M, C-134N, C-134O, C-134P, C-134Q, C-134R, C-134S, C-134T, C-134U, C-134V, C-134W, C-134X, C-134Y, C-134Z, C-135, C-135A, C-135B, C-135C, C-135D, C-135E, C-135F, C-135G, C-135H, C-135I, C-135J, C-135K, C-135L, C-135M, C-135N, C-135O, C-135P, C-135Q, C-135R, C-135S, C-135T, C-135U, C-135V, C-135W, C-135X, C-135Y, C-135Z, C-136, C-136A, C-136B, C-136C, C-136D, C-136E, C-136F, C-136G, C-136H, C-136I, C-136J, C-136K, C-136L, C-136M, C-136N, C-136O, C-136P, C-136Q, C-136R, C-136S, C-136T, C-136U, C-136V, C-136W, C-136X, C-136Y, C-136Z, C-137, C-137A, C-137B, C-137C, C-137D, C-137E, C-137F, C-137G, C-137H, C-137I, C-137J, C-137K, C-137L, C-137M, C-137N, C-137O, C-137P, C-137Q, C-137R, C-137S, C-137T, C-137U, C-137V, C-137W, C-137X, C-137Y, C-137Z, C-138, C-138A, C-138B, C-138C, C-138D, C-138E, C-138F, C-138G, C-138H, C-138I, C-138J, C-138K, C-138L, C-138M, C-138N, C-138O, C-138P, C-138Q, C-138R, C-138S, C-138T, C-138U, C-138V, C-138W, C-138X, C-138Y, C-138Z, C-139, C-139A, C-139B, C-139C, C-139D, C-139E, C-139F, C-139G, C-139H, C-139I, C-139J, C-139K, C-139L, C-139M, C-139N, C-139O, C-139P, C-139Q, C-139R, C-139S, C-139T, C-139U, C-139V, C-139W, C-139X, C-139Y, C-139Z, C-140, C-140A, C-140B, C-140C, C-140D, C-140E, C-140F, C-140G, C-140H, C-140I, C-140J, C-140K, C-140L, C-140M, C-140N, C-140O, C-140P, C-140Q, C-140R, C-140S, C-140T, C-140U, C-140V, C-140W, C-140X, C-140Y, C-140Z, C-141, C-141A, C-141B, C-141C, C-141D, C-141E, C-141F, C-141G, C-141H, C-141I, C-141J, C-141K, C-141L, C-141M, C-141N, C-141O, C-141P, C-141Q, C-141R, C-141S, C-141T, C-141U, C-141V, C-141W, C-141X, C-141Y, C-141Z, C-142, C-142A, C-142B, C-142C, C-142D, C-142E, C-142F, C-142G, C-142H, C-142I, C-142J, C-142K, C-142L, C-142M, C-142N, C-142O, C-142P, C-142Q, C-142R, C-142S, C-142T, C-142U, C-142V, C-142W, C-142X, C-142Y, C-142Z, C-143, C-143A, C-143B, C-143C, C-143D, C-143E, C-143F, C-143G, C-143H, C-143I, C-143J, C-143K, C-143L, C-143M, C-143N, C-143O, C-143P, C-143Q, C-143R, C-143S, C-143T, C-143U, C-143V, C-143W, C-143X, C-143Y, C-143Z, C-144, C-144A, C-144B, C-144C, C-144D, C-144E, C-144F, C-144G, C-144H, C-144I, C-144J, C-144K, C-144L, C-144M, C-144N, C-144O, C-144P, C-144Q, C-144R, C-144S, C-144T, C-144U, C-144V, C-144W, C-144X, C-144Y, C-144Z, C-145, C-145A, C-145B, C-145C, C-145D, C-145E, C-145F, C-145G, C-145H, C-145I, C-145J, C-145K, C-145L, C-145M, C-145N, C-145O, C-145P, C-145Q, C-145R, C-145S, C-145T, C-145U, C-145V, C-145W, C-145X, C-145Y, C-145Z, C-146, C-146A, C-146B, C-146C, C-146D, C-146E, C-146F, C-146G, C-146H, C-146I, C-146J, C-146K, C-146L, C-146M, C-146N, C-146O, C-146P, C-146Q, C-146R, C-146S, C-146T, C-146U, C-146V, C-146W, C-146X, C-146Y, C-146Z, C-147, C-147A, C-147B, C-147C, C-147D, C-147E, C-147F, C-147G, C-147H, C-147I, C-147J, C-147K, C-147L, C-147M, C-147N, C-147O, C-147P, C-147Q, C-147R, C-147S, C-147T, C-147U, C-147V, C-147W, C-147X, C-147Y, C-147Z, C-148, C-148A, C-148B, C-148C, C-148D, C-148E, C-148F, C-148G, C-148H, C-148I, C-148J, C-148K, C-148L, C-148M, C-148N, C-148O, C-148P, C-148Q, C-148R, C-148S, C-148T, C-148U, C-148V, C-148W, C-148X, C-148Y, C-148Z, C-149, C-149A, C-149B, C-149C, C-149D, C-149E, C-149F, C-149G, C-149H, C-149I, C-149J, C-149K, C-149L, C-149M, C-149N, C-149O, C-149P, C-149Q, C-149R, C-149S, C-149T, C-149U, C-149V, C-149W, C-149X, C-149Y, C-149Z, C-150, C-150A, C-150B, C-150C, C-150D, C-150E, C-150F, C-150G, C-150H, C-150I, C-150J, C-150K, C-150L, C-150M, C-150N, C-150O, C-150P, C-150Q, C-150R, C-150S, C-150T, C-150U, C-150V, C-150W, C-150X, C-150Y, C-150Z, C-151, C-151A, C-151B, C-151C, C-151D, C-151E, C-151F, C-151G, C-151H, C-151I, C-151J, C-151K, C-151L, C-151M, C-151N, C-151O, C-151P, C-151Q, C-151R, C-151S, C-151T, C-151U, C-151V, C-151W, C-151X, C-151Y, C-151Z, C-152, C-152A, C-152B, C-152C, C-152D, C-152E, C-152F, C-152G, C-152H, C-152I, C-152J, C-152K, C-152L, C-152M, C-152N, C-152O, C-152P, C-152Q, C-152R, C-152S, C-152T, C-152U, C-152V, C-152W, C-152X, C-152Y, C-152Z, C-153, C-153A, C-153B, C-153C, C-153D, C-153E, C-153F, C-153G, C-153H, C-153I, C-153J, C-153K, C-153L, C-153M, C-153N, C-153O, C-153P, C-153Q, C-153R, C-153S, C-153T, C-153U, C-153V, C-153W, C-153X, C-153Y, C-153Z, C-154, C-154A, C-154B, C-154C, C-154D, C-154E, C-154F, C-154G, C-154H, C-154I, C-154J, C-154K, C-154L, C-154M, C-154N, C-154O, C-154P, C-154Q, C-154R, C-154S, C-154T, C-154U, C-154V, C-154W, C-154X, C-154Y, C-154Z, C-155, C-155A, C-155B, C-155C, C-155D, C-155E, C-155F, C-155G, C-155H, C-155I, C-155J, C-155K, C-155L, C-155M, C-155N, C-155O, C-155P, C-155Q, C-155R, C-155S, C-155T, C-155U, C-155V, C-155W, C-155X, C-155Y, C-155Z, C-156, C-156A, C-156B, C-156C, C-156D, C-156E, C-156F, C-156G, C-156H, C-156I, C-156J, C-156K, C-156L, C-156M, C-156N, C-156O, C-156P, C-156Q, C-156R, C-156S, C-156T, C-156U, C-156V, C-156W, C-156X, C-156Y, C-156Z, C-157, C-157A, C-157B, C-157C, C-157D, C-157E, C-157F, C-157G, C-157H, C-157I, C-157J, C-157K, C-157L, C-157M, C-157N, C-157O, C-157P, C-157Q, C-157R, C-157S, C-157T, C-157U, C-157V, C-157W, C-157X, C-157Y, C-157Z, C-158, C-158A, C-158B, C-158C, C-158D, C-158E, C-158F, C-158G, C-158H, C-158I, C-158J, C-158K, C-158L, C-158M, C-158N, C-158O, C-158P, C-158Q, C-158R, C-158S, C-158T, C-158U, C-158V, C-158W, C-158X, C-158Y, C-158Z, C-159, C-159A, C-159B, C-159C, C-159D, C-159E, C-159F, C-159G, C-159H, C-159I, C-159J, C-159K, C-159L, C-159M, C-159N, C-159O, C-159P, C-159Q, C-159R, C-159S, C-159T, C-159U, C-159V, C-159W, C-159X, C-159Y, C-159Z, C-160, C-160A, C-160B, C-160C, C-160D, C-160E, C-160F, C-160G, C-160H, C-160I, C-160J, C-160K, C-160L, C-160M, C-160N, C-160O, C-160P, C-160Q, C-160R, C-160S, C-160T, C-160U, C-160V, C-160W, C-160X, C-160Y, C-160Z, C-161, C-161A, C-161B, C-161C, C-161D, C-161E, C-161F, C-161G, C-161H, C-161I, C-161J, C-161K, C-161L, C-161M, C-161N, C-161O, C-161P, C-161Q, C-161R, C-161S, C-161T, C-161U, C-161V, C-161W, C-161X, C-161Y, C-161Z, C-162, C-162A, C-162B, C-162C, C-162D, C-162E, C-162F, C-162G, C-162H, C-162I, C-162J, C-162K, C-162L, C-162M, C-162N, C-162O, C-162P, C-162Q, C-162R, C-162S, C-162T, C-162U, C-162V, C-162W, C-162X, C-162Y, C-162Z, C-163, C-163A, C-163B, C-163C, C-163D, C-163E, C-163F, C-163G, C-163H, C-163I, C-163J, C-163K, C-163L, C-163M, C-163N, C-163O, C-163P, C-163Q, C-163R, C-163S, C-163T, C-163U, C-163V, C-163W, C-163X, C-163Y, C-163Z, C-164, C-164A, C-164B, C-164C, C-164D, C-164E, C-164F, C-164G, C-164H, C-164I, C-164J, C-164K, C-164L, C-164M, C-164N, C-164O, C-164P, C-164Q, C-164R, C-164S, C-164T, C-164U, C-164V, C-164W, C-164X, C-164Y, C-164Z, C-165, C-165A, C-165B, C-165C, C-165D, C-165E, C-165F, C-165G, C-165H, C-165I, C-165J, C-165K, C-165L, C-165M, C-165N, C-165O, C-165P, C-165Q, C-165R, C-165S, C-165T, C-165U, C-165V, C-165W, C-165X, C-165Y, C-165Z, C-166, C-166A, C-166B, C-166C, C-166D, C-166E, C-166F, C-166G, C-166H, C-166I, C-166J, C-166K, C-166L, C-166M, C-166N, C-166O, C-166P, C-166Q, C-166R, C-166S, C-166T, C-166U, C-166V, C-166W, C-166X, C-166Y, C-166Z, C-167, C-167A, C-167B, C-167C, C-167D, C-167E, C-167F, C-167G, C-167H, C-167I, C-167J, C-167K, C-167L, C-167M, C-167N, C-167O, C-167P, C-167Q, C-167R, C-167S, C-167T, C-167U, C-167V, C-167W, C-167X, C-167Y, C-167Z, C-168, C-168A, C-168B, C-168C, C-168D, C-168E, C-168F, C-168G, C-168H, C-168I, C-168J, C-168K, C-168L, C-168M, C-168N, C-168O, C-168P, C-168Q, C-168R, C-168S, C-168T, C-168U, C-168V, C-168W, C-168X, C-168Y, C-168Z, C-169, C-169A, C-169B, C-169C, C-169D, C-169E, C-169F, C-169G, C-169H, C-169I, C-169J, C-169K, C-169L, C-169M, C-169N, C-169O, C-169P, C-169Q, C-169R, C-169S, C-169T, C-169U, C-169V, C-169W, C-169X, C-169Y, C-169Z, C-170, C-170A, C-170B, C-170C, C-170D, C-170E, C-170F, C-170G, C-170H, C-170I, C-170J, C-170K, C-170L, C-170M, C-170N, C-170O, C-170P, C-170Q, C-170R, C-170S, C-170T, C-170U, C-170V, C-170W, C-170X, C-170Y, C-170Z, C-171, C-171A, C-171B, C-171C, C-171D, C-171E, C-171F, C-171G, C-171H, C-171I, C-171J, C-171K, C-171L, C-171M, C-171N, C-171O, C-171P, C-171Q, C-171R, C-171S, C-171T, C-171U, C-171V, C-171W, C-171X, C-171Y, C-171Z, C-172, C-172A, C-172B, C-172C, C-172D, C-172E, C-172F, C-172G, C-172H, C-172I, C-172J, C-172K, C-172L, C-172M, C-172N, C-172O, C-172P, C-172Q, C-172R, C-172S, C-172T, C-172U, C-172V, C-172W, C-172X, C-172Y, C-172Z, C-173, C-173A, C-173B, C-173C, C-173D, C-173E, C-173F, C-173G, C-173H, C-173I, C-173J, C-173K, C-173L, C-173M, C-173N, C-173O, C-173P, C-173Q, C-173R, C-173S, C-173T, C-173U, C-173V, C-173W, C-173X, C-173Y, C-173Z, C-174, C-174A, C-174B, C-174C, C-174D, C-174E, C-174F, C-174G, C-174H, C-174I, C-174J, C-174K, C-174L, C-174M, C-174N, C-174O, C-174P, C-174Q, C-174R, C-174S, C-174T, C-174U, C-174V, C-174W, C-174X, C-174Y, C-174Z, C-175, C-175A, C-175B, C-175C, C-175D, C-175E, C-175F, C-175G, C-175H, C-175I, C-175J, C-175K, C-175L, C-175M, C-175N, C-175O, C-175P, C-175Q, C-175R, C-175S, C-175T, C-175U, C-175V, C-175W, C-175X, C-175Y, C-175Z, C-176, C-176A, C-176B, C-176C, C-176D, C-176E, C-176F, C-176G, C-176H, C-176I, C-176J, C-176K, C-176L, C-176M, C-176N, C-176O, C-176P, C-176Q, C-176R, C-176S, C-176T, C-176U, C-176V, C-176W, C-176X, C-176Y, C-176Z, C-177, C-177A, C-177B, C-177C, C-177D, C-177E, C-177F, C-177G, C-177H, C-177I, C-177J, C-177K, C-177L, C-177M, C-177N, C-177O, C-177P, C-177Q, C-177R, C-177S, C-177T, C-177U, C-177V, C-177W, C-177X, C-177Y, C-177Z, C-178, C-178A, C-178B, C-178C, C-178D, C-178E, C-178F, C-178G, C-178H, C-178I, C-178J, C-178K, C-178L, C-178M, C-178N, C-178O, C-178P, C-178Q, C-178R, C-178S, C-178T, C-178U, C-178V, C-178W, C-178X, C-178Y, C-178Z, C-179, C-179A, C-179B, C-179C, C-179D, C-179E, C-179F, C-179G, C-179H, C-179I, C-179J, C-179K, C-179L, C-179M, C-179N, C-179O, C-179P, C-179Q, C-179R, C-179S, C-179T, C-179U, C-179V, C-179W, C-179X, C-179Y, C-179Z, C-180, C-180A, C-180B, C-180C, C-180D, C-180E, C-180F, C-180G, C-180H, C-180I, C-180J, C-180K, C-180L, C-180M, C-

Emlyn Hughes



INTERNATIONAL



**GAME
OF THE YEAR!**

**WINNER OF CCI OSKARS FOR
BEST SPORTS SIMULATION
AND BEST GAME OF THE YEAR!**

*Quite simply the best football
simulation available for the 64
Nothing short of superb*

ZZAP 64

**AVAILABLE NOW FOR
COMMODORE 64
£9.95 tape £12.95 disk**
Coming soon for Spectrum & Amstrad

Audiogenic

Audiogenic Software Limited, Winchester House,
Canning Road, Harrow HA3 7SL, England

Order by phone on 01 863 1166

